

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



Grado en Ingeniería Informática y Matemáticas

TRABAJO FIN DE GRADO

Comunicador para diferentes plataformas Android

Guillermo Díaz Gil
Tutor: Eduardo Boemo Scalvinoni

Julio 2020

Comunicador para diferentes plataformas Android

AUTOR: Guillermo Díaz Gil
TUTOR: Eduardo Boemo Scalvinoni

Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2020

Resumen (castellano)

Los dispositivos móviles, ya sean *Smartphone* o *Tablets* son herramientas que utilizamos a diario y por tanto es coherente pensar en ellas como herramientas en aulas escolares, trabajos, etc.

Android es el sistema Operativo para dispositivos móviles que mayor crecimiento experimenta en los últimos años. Y por esto su adaptación en muchos ámbitos es importante.

El presente proyecto viene a cubrir un ámbito poco utilizado para este tipo de dispositivos. Ser una herramienta de ayuda en la comunicación de personas con discapacidad, principalmente motoras, que les impida utilizar los dispositivos de forma usual, es decir con las manos.

Este proyecto se enfoca en ofrecer una aplicación muy sencilla que ayude a estas personas a comunicarse en base a unos pictogramas básicos y movimientos de cabeza para controlar la aplicación. El objetivo principal radica en la simpleza de la interfaz, para que sea todo lo más intuitivo para el usuario y fácilmente se convierta en una herramienta que use a diario.

A lo largo del documento se presenta la aplicación realizada, exponiendo un estudio de su antecesora, una descripción de características y requisitos principales, todas las tecnologías utilizadas en el desarrollo de la aplicación, un análisis de todas las fases del proyecto, incluyendo una fase de pruebas y por último un manual de uso para una mejor comprensión y facilidad de instalación.

Este último punto, el manual, es muy importante en este proyecto debido a que el público de esta aplicación serán profesores, padres, etc. Que trabajen con personas discapacitadas y requieran de esta herramienta para facilitar la comunicación entre ellos.

Abstract (English)

Mobile devices, such as *Smartphones* or *Tablets* are tools used every day, that is why it is coherent to think in them as tools in schools and jobs.

Android is the Operative System used in mobile devices which has experienced the greatest growth lately. Its adaptation in different fields is of extreme importance.

The following project tries to cover a field that has not been used much with this type of devices. Being a tool to help people with special needs in communication. Specially for people with motoric problems, with difficulties to use these devices using their hands.

This project is focused in offering a very easy application to help these people to communicate with basic pictograms and head movements to control the application. The main objective is the simplicity of its interface, being as intuitive as possible to the user and making it easy to be used as a daily tool.

This document shows the application, researching the previous ones. A description of its characteristics and its main features. All the technologies used in the development of the

application. It analyses all the stages of the project, including a trial period and finally a user's manual to facilitate installation and comprehension.

This last point: the user's manual, is very important in this project as the receivers of this application are parents and teachers, people in charge of working with those with special needs and requiring this tool to make communication easier between them.

Palabras clave

La siguiente lista contiene palabras descriptivas que podrán utilizarse para catalogar el proyecto.

- Android.
- Aplicación.
- Smartphone.
- Tablet.
- Personas con discapacidad motora.

Keywords (inglés)

The following list contains descriptive words that can be used to catalog the project.

- Android.
- Application.
- Smartphone.
- Tablet.
- People with special needs: motoric problems.

Agradecimientos

“A mis padres, mi hermana y mi mejor amiga Cris, gracias por todo el apoyo, energía y fuerza que me han dado transmitido durante toda mi carrera universitaria”

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	1
1.3	Fases de realización.....	2
1.4	Organización de la memoria.....	2
2	Estado del arte	3
2.1	Quiero Decirte... ..	3
2.2	Comunica de Irisbond.....	4
2.3	Speak for Yourself.....	5
2.4	CaughDrop AAC	5
3	EVA Facial Mouse	7
3.1	¿Qué es?.....	7
3.2	Limitaciones	7
4	Análisis	9
4.1	Catálogo de requisitos	9
4.1.1	Requisitos funcionales.....	9
4.1.2	Requisitos no funcionales.....	10
4.2	Base de datos	10
5	Diseño.....	11
5.1	General.....	11
5.2	Tecnologías utilizadas	11
5.2.1	NinjaMocks	11
5.2.2	Visual Paradigm	12
5.2.3	Android Studio	12
5.2.4	Android SDK.....	12
5.3	Diseño de la interfaz	12
5.3.1	Pantalla de Inicio	13
5.3.2	Pantalla categorías	14
5.3.3	Pantalla de Ajustes	15
5.4	Diseño de la base de datos.....	15
6	Desarrollo	17
6.1	General.....	17
6.1.1	Modelo.....	17
6.1.1.1	Base de datos	17
6.1.1.2	Monitor	17
6.1.2	Vista.....	18
6.1.3	Controlador.....	18
6.1.3.1	MainActivity.....	18
6.1.3.2	PictogramaActivity	21
6.1.3.3	SettingActivity.....	22
7	Pruebas	24
7.1	Plan de pruebas	24
8	Conclusiones y trabajo futuro.....	26
8.1	Conclusiones.....	26
8.2	Trabajo futuro	26
	Referencias	29
	Glosario	31
	Anexos.....	I
A	Manual de instalación.....	I

A1.	EVA Facial Mouse	I
A2.	TFG.....	XIII
B	Carta CEIP Romero Peña de La Solana	XV

INDICE DE FIGURAS

FIGURA 2-1 QUIERO DECIRTE... CAPTURA 1	3
FIGURA 2-2 QUIERO DECIRTE... CAPTURA 2	4
FIGURA 2-3 SPREAK FOR YOURSELF	5
FIGURA 2-4 COUGHDROP AAC.....	6
FIGURA 5-1 PROTOTIPO PANTALLA DE INICIO	13
FIGURA 5-2 PROTOTIPO PANTALLA DE CATEGORÍAS	14
FIGURA 5-3 PROTOTIPO PANTALLA AJUSTES	15
FIGURA 5-4 MODELO ENTIDAD RELACIÓN	16
FIGURA 5-5 TABLAS DE LA BASE DE DATOS.....	16
FIGURA 6-1 RECEIVER ANDORIDMANIFIEST	18
FIGURA 6-2 PERMISOS ANDROIDMANIFIEST	18
FIGURA 6-3 PANTALLA PRINCIPAL PORTRAIT.....	19
FIGURA 6-4 PANTALLA PRINCIPAL LANDSCAPE	19
FIGURA 6-5 CÓDIGO PANTALLA COMPLETA.....	20
FIGURA 6-6 PANTALLA PICTOGRAMA PORTRAIT.....	21
FIGURA 6-7 PANTALLA PICTOGRAMA LANDSCAPE	22
FIGURA 6-8 PANTALLA <i>SETTINGS</i>	23
FIGURA 0-1 INSTALACIÓN EVA FACIAL MOUSE 1	I
FIGURA 0-2 INSTALACIÓN EVA FACIAL MOUSE 2	II
FIGURA 0-3 INSTALACIÓN EVA FACIAL MOUSE 3	II
FIGURA 0-4 INSTALACIÓN EVA FACIAL MOUSE 4	III
FIGURA 0-5 INSTALACIÓN EVA FACIAL MOUSE 5	IV
FIGURA 0-6INSTALACIÓN EVA FACIAL MOUSE 4	IV
FIGURA 0-7 INSTALACIÓN EVA FACIAL MOUSE 7	V

FIGURA 0-8 INSTALACIÓN EVA FACIAL MOUSE 8	V
FIGURA 0-9 INSTALACIÓN EVA FACIAL MOUSE 9	VI
FIGURA 0-10 INSTALACIÓN EVA FACIAL MOUSE 10	VI
FIGURA 0-11INSTALACIÓN EVA FACIAL MOUSE 11	VII
FIGURA 0-12 INSTALACIÓN EVA FACIAL MOUSE 12	VIII
FIGURA 0-13 INSTALACIÓN EVA FACIAL MOUSE 13	VIII
FIGURA 0-14 INSTALACIÓN EVA FACIAL MOUSE 14	IX
FIGURA 0-15 INSTALACIÓN EVA FACIAL MOUSE 15	IX
FIGURA 0-16 INSTALACIÓN EVA FACIAL MOUSE 16	X
FIGURA 0-17 INSTALACIÓN EVA FACIAL MOUSE 17	X
FIGURA 0-18 INSTALACIÓN EVA FACIAL MOUSE 18	XI
FIGURA 0-19 INSTALACIÓN EVA FACIAL MOUSE 19	XI
FIGURA 0-20 INSTALACIÓN EVA FACIAL MOUSE 20	XII
FIGURA 0-21 INSTALACIÓN EVA FACIAL MOUSE 21	XII
FIGURA 0-22 INSTALACIÓN EVA FACIAL MOUSE 22	XIII

1 Introducción

1.1 Motivación

Este trabajo de fin de grado nace con el objetivo de ayudar a personas con discapacidades o problemas de lenguaje, niños no verbales, autismo, parálisis cerebral, deficiencia intelectual o cualquier otro trastorno generalizado del desarrollo con un nivel cognitivo muy bajo.

Con anterioridad a este trabajo encontramos la aplicación “Quiero decirte...” [1] desarrollada por DSLab UAM. Esta aplicación es un comunicador para *Tablets* y teléfonos Android que mediante botones, gestos y pictogramas de ARASAAC [2], (Portal aragonés de la Comunicación Aumentativa y Alternativa), facilita y potencia la comunicación espontánea mediante imágenes y sonidos, permitiendo desarrollar comportamientos interactivos espontáneos intencionales.

Tras dicha aplicación se decide crear este proyecto, basado en una segunda aplicación que mediante la APP de Vodafone “EVA Facial Mouse” [3] se pueda controlar utilizando la cabeza como puntero y así seleccionar los pictogramas sin la necesidad de tocar la pantalla.

Por lo tanto, esta app se centra más en personas con discapacidad motora en las manos.

1.2 Objetivos

El principal objetivo es crear una aplicación que sea funcional y pueda ser utilizada por gente con discapacidad motora.

En este Trabajo de Fin de Grado se realiza una primera versión de esta aplicación cuyo objetivo es desarrollar una aplicación que se pueda llevar al CEIP Romero Peña, colegio público de Ciudad Real colaborador con DSLab UAM para el desarrollo de esta aplicación y su antecesora.

Los objetivos específicos de la aplicación a desarrollar son:

- Crear una aplicación con un diseño sencillo e intuitivo para ello se utilizan pictogramas de ARASAAC [2], dichos pictogramas son imágenes sencillas con poco color con la finalidad de crear pocos estímulos al usuario y ser identificables.
- Manual de usuario.
- *Booting* directa.
- No permitir acceso a otras aplicaciones.
- Cambio entre categorías.
- Volver al inicio de la aplicación.
- Pulsar los pictogramas mediante fijación del puntero con la cabeza.

1.3 Fases de realización

En este apartado vamos a dar una visión general de las fases seguidas a lo largo de la realización del proyecto, que son las correspondientes a la mayoría de un desarrollo de software.

- **Estudio:** Se ha llevado a cabo una investigación sobre posibles tecnologías a utilizar para la realización del proyecto y un estudio de mercado.
- **Análisis:** Definición de requisitos que se quieren conseguir en el proyecto.
- **Diseño:** Tras las fases de estudio y análisis, se ha procedido a diseñar la arquitectura de la aplicación.
Aquí se incluyen diseños de pantallas, interfaz y base de datos.
- **Desarrollo:** Programación de la aplicación Android en Java.
- **Pruebas:** Una vez finalizadas las fases anteriores, se ha realizado una fase de pruebas general para comprobar la calidad y efectividad del software desarrollado.
- **Documentación:** Una parte muy importante de este proyecto es el manual de usuario. Éste tiene que estar muy claro y ayudar de forma sencilla a instalar y comprender la aplicación.
- **Continuación:** Se realiza un breve análisis de posibles formas de continuar y objetivos futuros para el proyecto.

1.4 Organización de la memoria

El presente documento recoge todas las fases mencionadas anteriormente con la siguiente estructura:

- Introducción.
- Estado del arte.
- EVA Facial Mouse [3].
- Análisis.
- Diseño.
- Desarrollo.
- Pruebas.
- Conclusión y trabajos futuros.

Así como un manual de uso añadido en los anexos, con el cual cualquier usuario podría instalar, aprender y utilizar la aplicación fácilmente.

2 Estado del arte

2.1 Quiero Decirte...

“Quiero Decirte...” [1] es una aplicación desarrollada por DSLab UAM como un comunicador para *Tablets* y teléfonos Android que, mediante botones, gestos y pictogramas, facilita y potencia la comunicación espontánea mediante imágenes y sonidos, permitiendo desarrollar comportamientos interactivos espontáneos e intencionales.

Dicha aplicación se puede encontrar en *Google Play* para su instalación y uso. Además de estar en constante actualización (última actualización a 21 de marzo de 2019).

A continuación, vamos a ver varias capturas de pantalla de la aplicación, para tener una idea de cómo es y cómo funciona.

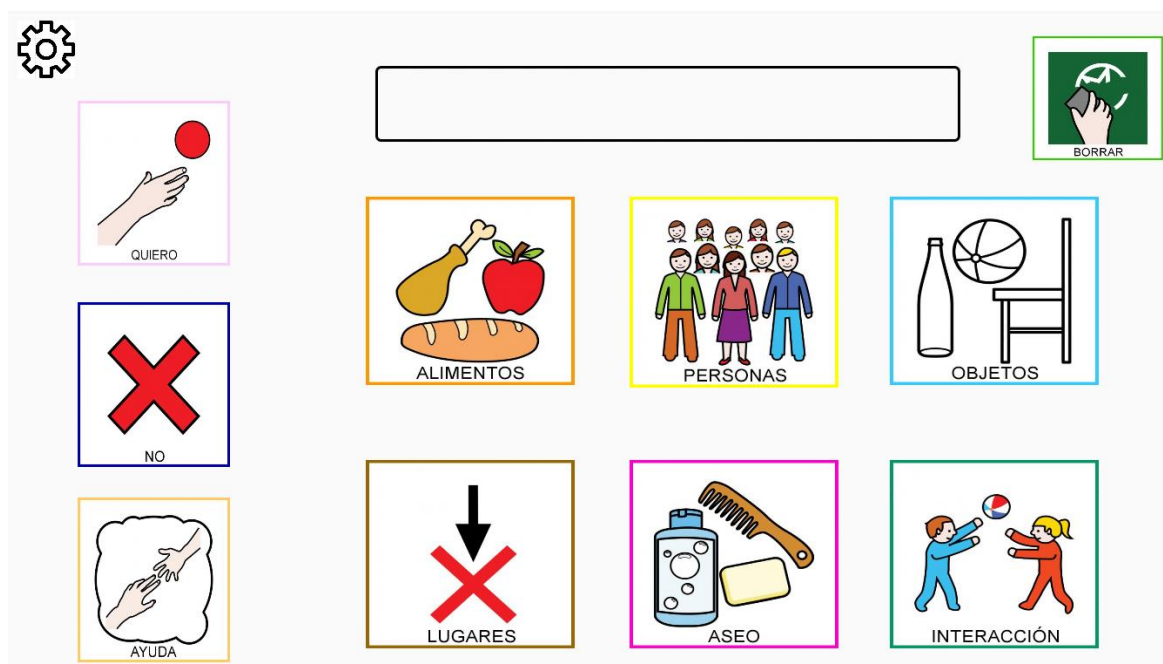


Figura 2-1 Quiero Decirte... captura 1

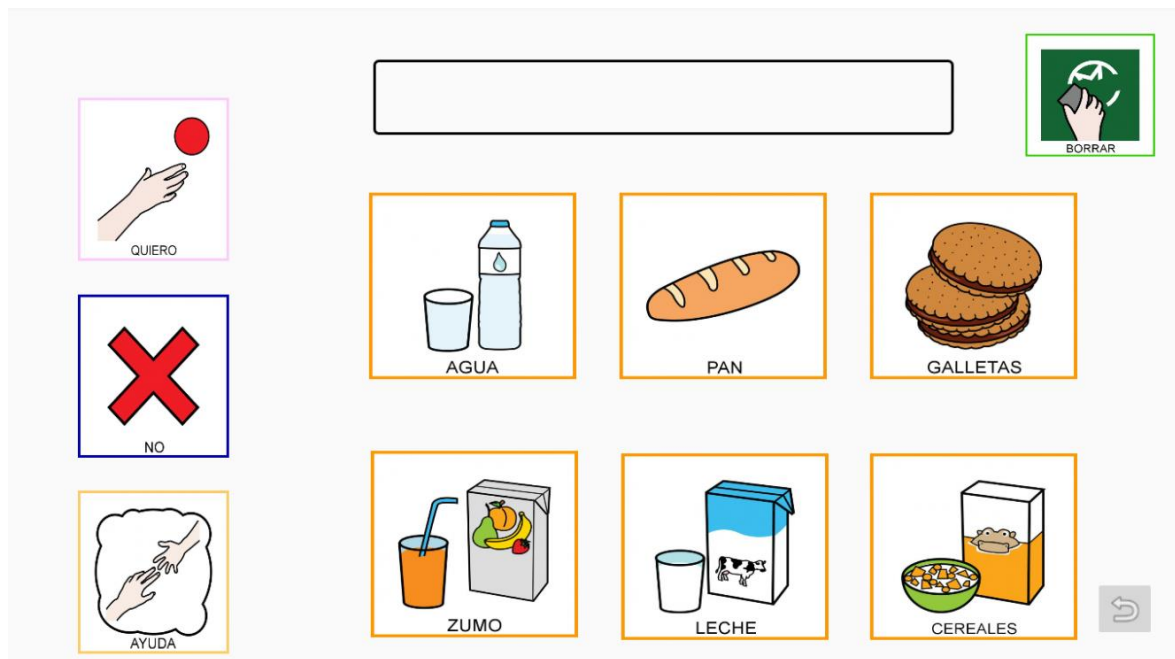


Figura 2-2 Quiero Decirte... captura 2

Cómo podemos ver en las imágenes tenemos una interfaz sencilla con los diferentes pictogramas que podemos utilizar.

En la primera imagen vemos la pantalla con las categorías y en la segunda con los elementos de esa categoría.

Además, encontramos tres pictogramas fijos: quiero, no y ayuda para un acceso más rápido. En la segunda imagen también hay un botón para retroceder a la pantalla de categorías.

2.2 Comunica de Irisbond

Irisbond [4] es una empresa que desde 2013 desarrolla tecnología con impacto social. Está enfocado a personas con discapacidades motoras en silla de ruedas. Actualmente cuenta con dos proyectos: Aprende y Comunica. Ambos proyectos tienen como base un lector ocular de alta precisión que permite interactuar y controlar el ordenador de manera sencilla, utilizando un ratón tradicional, pero con los ojos, gracias a un sistema basado en infrarrojos, Irisbond Duo.

Aprende es una solución para cubrir la curva de aprendizaje de los usuarios: personas con síndrome Rett, personas con parálisis cerebral, personas con posibles afecciones cognitivas, personas con lesión cerebral adquirida y centros especializados dedicados a trabajar con personas que tengan estas afecciones.

Comunica es una solución para cubrir la necesidad de expresarse libremente y de manera totalmente autónoma destinada a: personas afectadas por ELA, personas con tetraplejia, personas con lesión medular espinal, personas con altas capacidades cognitivas impedidas

físicamente o con trastornos del habla, personas con parálisis cerebral y centros especializados dedicados a trabajar con personas que tengan estas afecciones.

Comunica es la única solución que se ha encontrado en la actualidad para ayudar a personas con estas afecciones en su comunicación. Pero, su elevado precio imposibilita su adquisición a la mayoría de los usuarios a los que está destinado. Siendo el precio de Irisbond Duo de 1.600€ y de Comunica la solución completa de 4.236,1€ la versión más económica.

2.3 Speak for Yourself

Speak for Yourself [5] es una aplicación para IOS diseñada para la comunicación Aumentativa y alternativa. Esta aplicación al igual que “Quiero decirte...” se basa en la manipulación táctil del dispositivo para la ayuda en la comunicación, en este caso teniendo una interfaz con botones mucho más pequeños.



Figura 2-3 Speak for Yourself

Debido a su interfaz no es adecuada para las personas con las afecciones a las que está destinado nuestro proyecto.

El precio de esta aplicación es de 299,99\$ y se puede encontrar en la *App Store* de Apple tanto para *iPhone* como para *iPad*.

2.4 CoughDrop AAC

CoughDrop AAC [6] es una herramienta de comunicación y soporte AAC simple para personas con autismo, parálisis cerebral, síndrome de Down, síndrome de Angelman, síndrome de Rett, etc. Se basa en la unión de pictogramas para formular frases.

Para su funcionamiento necesitas tener una cuenta en coughdrop.com y pagar la licencia de 6\$/mes o 200\$ en pago único.

Esta aplicación se puede encontrar en *App Store*, *Google Play*, *Amazon*, *Windows* y en *Web Browser*.

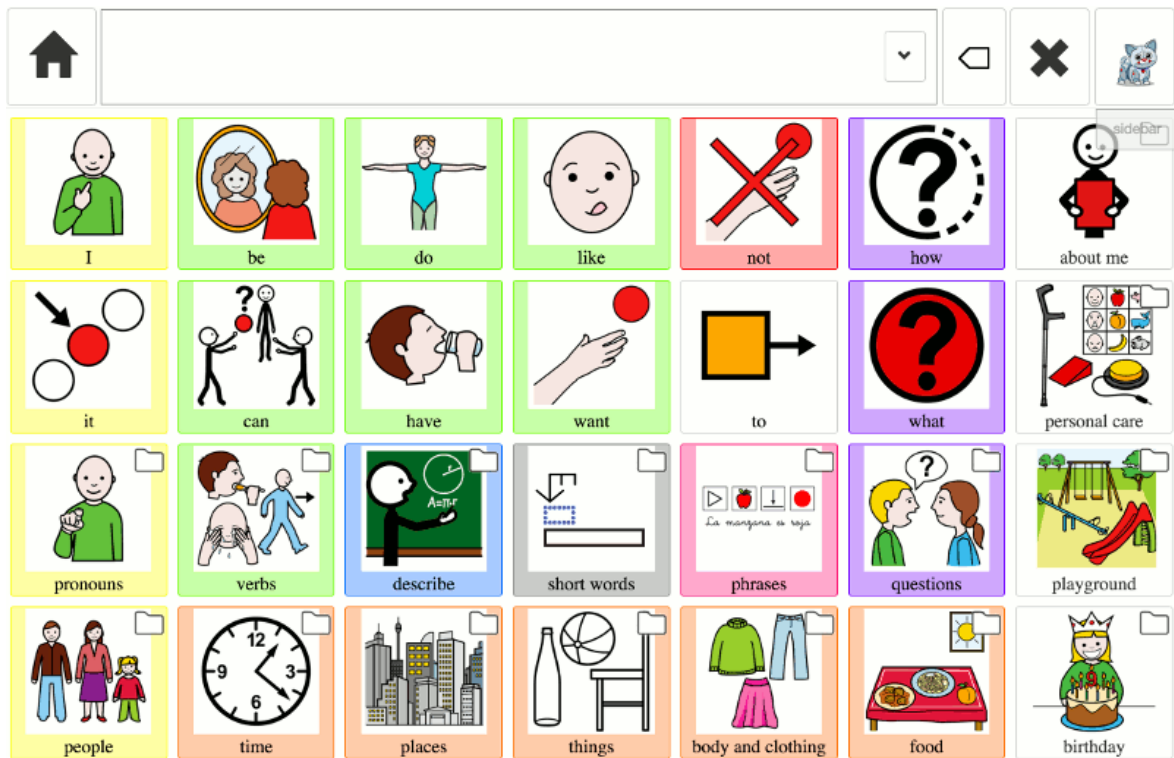


Figura 2-4 CoughDrop AAC

3 EVA Facial Mouse

3.1 ¿Qué es?

EVA Facial Mouse [3] es una aplicación gratuita y de código abierto que permite controlar de forma alternativa (manos libres) las funciones del dispositivo móvil por medio del seguimiento del rostro del usuario captado a través de la cámara frontal. A partir del movimiento del rostro permite controlar un puntero en pantalla (a modo de ratón) que proporciona el acceso directo a la mayor parte de elementos de la interfaz de usuario.

Alguno de los colectivos que pueden beneficiarse de este proyecto son las personas con amputaciones, parálisis cerebral, lesión medular, distrofia muscular, esclerosis múltiple, esclerosis lateral amiotrófica (ELA) y otras discapacidades.

Para el desarrollo de EVA Facial Mouse [3] se ha contado con el apoyo de la Fundación Vodafone España y la colaboración de expertos y usuarios de organizaciones de la Confederación ASPACE (España). A saber, ASPACE Asturias, ASPACE Barcelona, ASPACE Gipuzkoa, ASPACE Granada, APPC Tarragona y AVAPACE. Además, ha sido probado por CEAPAT, CRE (León) y ASPAYM Castilla y León.

La Fundación Vodafone España es una institución privada, sin ánimo de lucro, que tiene como objetivo mejorar la calidad vida de los colectivos vulnerables a través de las Tecnologías de la Información y las Comunicaciones. Colabora con diversas entidades, tanto públicas como privadas, en proyectos de innovación tecnológica y sus aplicaciones para la autonomía, la independencia y la integración socio-laboral de todas las personas.

3.2 Limitaciones

Para el uso de esta aplicación tenemos que tener en cuenta algunas limitaciones que nos comenta el equipo de desarrollo:

- No se puede usar simultáneamente con otras aplicaciones que hacen uso de la cámara.
- Para las versiones de Android anteriores a 5.1, la mayoría de los teclados estándar no funcionan con EVA, por lo que se proporciona un teclado básico. Dicho teclado debe activarse manualmente después de la instalación.
- No funciona con la mayoría de los juegos.
- Los navegadores no manejan correctamente algunas acciones (recomendamos utilizar Google Chrome).
- Aplicaciones como Maps, Earth y Gallery funcionan con restricciones.

Por motivos evidentes EVA Facial Mouse [3] no ha sido probada con todos os dispositivos existentes.

En su página de Git Hub [7] nos dan un listado de los dispositivos con los que ha sido probado. Dicho listado se puede modificar para añadir nuevos dispositivos. En mi caso he usado un dispositivo que no estaba en la lista y todo ha funcionado perfectamente.

También conocemos los requerimientos mínimos que necesita un dispositivo para poder instalar EVA Facial Mouse [3]:

- Teléfono móvil o tableta.
- Android 4.1 (Jelly Bean) o superior.
- Cámara frontal.
- Procesador de doble núcleo o superior.

4 Análisis

En este capítulo se presenta la fase de análisis, parte inicial de todo proyecto de software, se definen los requisitos y se muestra una vista global de la arquitectura pensada para el sistema.

Este capítulo es importante para asentar las bases del proyecto, a partir de las cuales se desarrollará todo lo demás.

Empezaremos por realizar una buena definición de los requisitos que debería tener el proyecto.

4.1 Catálogo de requisitos

El catálogo de requisitos es la especificación del comportamiento que se espera de cualquier proyecto de software. Estudiando otras aplicaciones similares, se ha predefinido una serie de requisitos que se consideran indispensables para el proyecto. Así como requisitos que se quieren incorporar una vez la base del proyecto este desarrollada.

4.1.1 Requisitos funcionales

Los requisitos funcionales describen todas las interacciones que tendrán los usuarios con el software.

RF1. Pantalla principal

La pantalla principal contará de un menú inferior con dos botones y un listado de categorías:

- a. Botón siguiente: para avanzar en la lista de categorías.
- b. Botón atrás: para retroceder en la lista de categorías.

RF2. Pantalla categorías

Esta pantalla contará con un menú inferior con tres botones y un listado de categorías.:

- a. Botón siguiente: para avanzar en la lista de categorías.
- b. Botón atrás: para retroceder en la lista de categorías.
- c. Botón inicio: para regresar a la pantalla principal.

RF3. Menú de ajustes

Dentro del menú de ajustes el usuario podrá controlar varios aspectos de la aplicación:

- a. *Booting* directa: el usuario podrá activar o desactivar la opción de *booting* directa de la aplicación al iniciar el dispositivo.
- b. Adición y modificación de pictogramas y categorías. (Ajuste futuro).

4.1.2 Requisitos no funcionales

Los requisitos no funcionales son requisitos complementarios o atributos de calidad. Especifican criterios que juzgan operaciones del sistema en lugar de su comportamiento.

RNF1. Documentación

- a. Manual de usuario de la aplicación.
- b. Manual del programador, en el que se incluirá documentación clara sobre la codificación de la aplicación.

RNF2. Mantenibilidad y portabilidad

- a. Disponibilidad para diferentes dispositivos Android (*Smartphone* o *Tablets*).
- b. La aplicación Android estará soportada para la mayoría de las versiones posibles.

RNF3. Interfaz y usabilidad

- a. La aplicación constará de una interfaz sencilla, atractiva e intuitiva. De forma que su uso no suponga un esfuerzo al usuario a la hora de utilizarla.
- b. Todos los accesos a la base de datos no supondrán demasiada carga para el dispositivo, por lo que el rendimiento será óptimo.

4.2 Base de datos

La base de datos será local en el propio dispositivo. Esta base de datos almacenará toda la información sobre los pictogramas utilizados y está en constante actualización dependiendo de las necesidades de los usuarios.

En el Capítulo 5 – Diseño se explicará con más detalle la base de datos.

5 Diseño

5.1 General

En este capítulo se describe todo el proceso de diseño de la aplicación. Se ha realizado un diseño que abarca todos los requisitos descritos en el apartado 4.1 Catálogo de requisitos. El diseño proporciona una idea completa del software desarrollado en el proyecto. Además, se justifican las decisiones tomadas para el posterior desarrollo.

Todo el diseño de la aplicación tiene que ser lo más simple posible, para ello nos basamos en pictogramas sencillos facilitados por ARASAAC [2], (Portal aragonés de la Comunicación Aumentativa y Alternativa).

Dichos pictogramas son imágenes sencillas con pocos estímulos, color y líneas. De esta forma son fácilmente reconocibles por los usuarios.

El diseño de la aplicación ha ido sufriendo algún cambio a lo largo de su desarrollo. En consecuencia, se muestran en este documento explicaciones de las versiones finales de diseño.

Uno de los cambios principales de diseño fue un menú inferior que consta de varios elementos para poder moverse entre pantallas, retroceder, avanzar, etc. Esto se debe a que los botones habituales de Android son físicos y el usuario no podría seleccionarlos con el cursor de la pantalla, recordemos que estamos ante una aplicación desarrollada para personas con discapacidad motora. Por otro lado, se pensó en solucionarlo con el menú de control virtual que se puede utilizar en Android y que muchos dispositivos utilizan por defecto. Pero uno de los requisitos que se quiere implementar es que el usuario no pueda salir de la aplicación y por lo tanto estos botones virtuales no nos servirían. Por lo tanto, se toma la decisión final de utilizar un menú inferior propio.

Dicho menú será explicado en cada una de las pantallas.

Otro aspecto importante de la aplicación era el tamaño de los iconos sobre todo en los Smartphone, y por eso y junto a la decisión de neutralizar el menú virtual de Android se ha decidido que la aplicación sea en pantalla completa.

5.2 Tecnologías utilizadas

En este pequeño apartado se enumeran las diferentes herramientas y tecnologías utilizadas durante el desarrollo del proyecto.

5.2.1 NinjaMocks

NinjaMocks [8] es un constructor gráfico de maquetas. Permite al diseñador organizar y diseñar las maquetas utilizando su potente herramienta *drag&drop*, con la que puedes arrastrar elementos y moverlos fácilmente dentro de la maqueta. Dentro de los numerosos

editores de maquetas existentes, NinjaMocks [8] permitía la creación de un proyecto para Android, lo que facilitaba el uso de los widgets que posee este sistema operativo.

5.2.2 Visual Paradigm

Visual Paradigm [9] es una herramienta de diseño para proyectos de software. Soporta multitud de estándares de modelado como UML, SysML, etc. Con esta herramienta hemos podido realizar diagramas para el diseño de nuestra aplicación.

5.2.3 Android Studio

Android Studio [10] es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan la productividad durante la compilación de apps para Android:

- Un sistema de compilación basado en *Gradle* flexible.
- Un emulador rápido con varias funciones.
- Un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android.
- *Instant Run* para aplicar cambios mientras tu app se ejecuta sin la necesidad de compilar un nuevo APK.
- Integración de plantillas de código y GitHub para ayudarte a compilar funciones comunes de las apps e importar ejemplos de código.
- Gran cantidad de herramientas y *frameworks* de prueba.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versión, etc.
- Compatibilidad con C++ y NDK.
- Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine.

5.2.4 Android SDK

El Android SDK [10] (Android Software Development Kit) contiene herramientas necesarias para el desarrollo Android y ejecutar el emulador del sistema Android en la versión que sea.

5.3 Diseño de la interfaz

La interfaz de esta aplicación se ha diseñado con el propósito de que sea intuitiva y muy simple.

En este apartado se ha utilizado NinjaMocks [8] para el diseño de las maquetas.

5.3.1 Pantalla de Inicio

La pantalla de inicio consta de un listado de categorías que hacen de división de los pictogramas. De esta forma, se facilita el acceso por grupos, además cada categoría está relacionada con un color para que sea más fácil de distinguir.

Las categorías iniciales son:

- Alimentos: naranja.
- Objetos: azul claro.
- Aseo: rosa.
- Interacción: verde.
- Lugares: marrón.
- Personas: amarillo.

Además de las categorías hay dos pictogramas más en la lista de la pantalla de inicio:

- No: azul oscuro.
- Ayuda: naranja.

Por último, se han añadido en la parte inferior de la pantalla dos flechas para poder ir avanzando o retrocediendo en la lista de categorías.

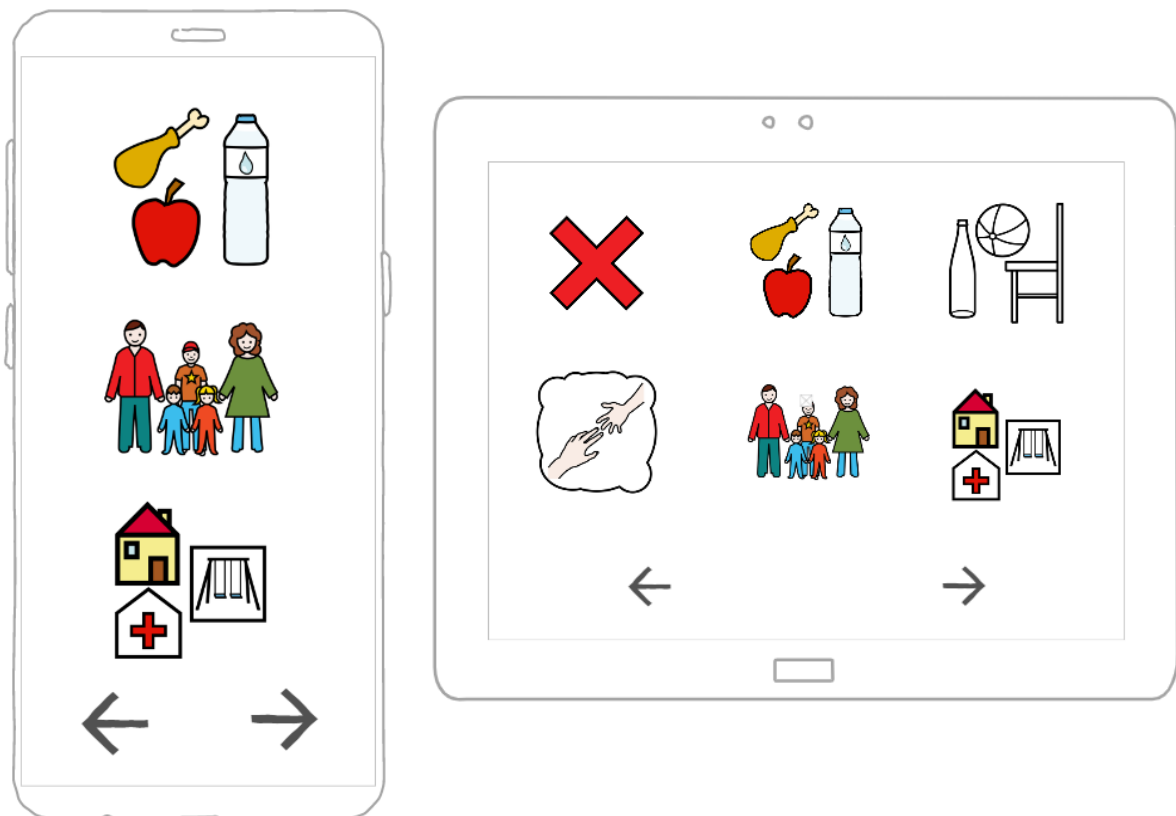


Figura 5-1 Prototipo Pantalla de Inicio

5.3.2 Pantalla categorías

Esta pantalla es específica para cada categoría y en ella se encuentran los pictogramas asociados a esa categoría.

La aplicación inicialmente tiene cinco pictogramas por cada categoría. Como hemos dicho en el apartado anterior cada categoría está relacionada con un color y por lo tanto los pictogramas de dicha categoría también lo están.

- Alimentos: agua, pan, galletas, zumo y leche.
- Objetos: tele, pelota, puzle, coche y cuento.
- Aseo: lavar los dientes, duchar, peinar, lavar las manos y hacer pis.
- Interacción: abrazo, cosquillas, besos, caricia y jugar.
- Lugares: parque, casa, colegio, piscina y recreo.
- Personas: papá, mamá, hermano, abuela y maestra.

En estas pantallas también se han añadido en la parte inferior dos flechas para poder avanzar o retroceder en la lista de pictogramas.

Pero, en este caso también se ha añadido un botón para regresar a la pantalla de inicio.



Figura 5-2 Prototipo Pantalla de Categorías

5.3.3 Pantalla de Ajustes

Esta pantalla se ha diseñado con intención de mejorarla y añadir más funcionalidades en un futuro, pero actualmente en esta primera versión de la aplicación es muy sencilla y solo nos dejara decidir si queremos *booting* directa.

Contará de una primera pantalla para decidir a qué ajustes queremos dirigirnos y otra con la lista de ajustes seleccionado.

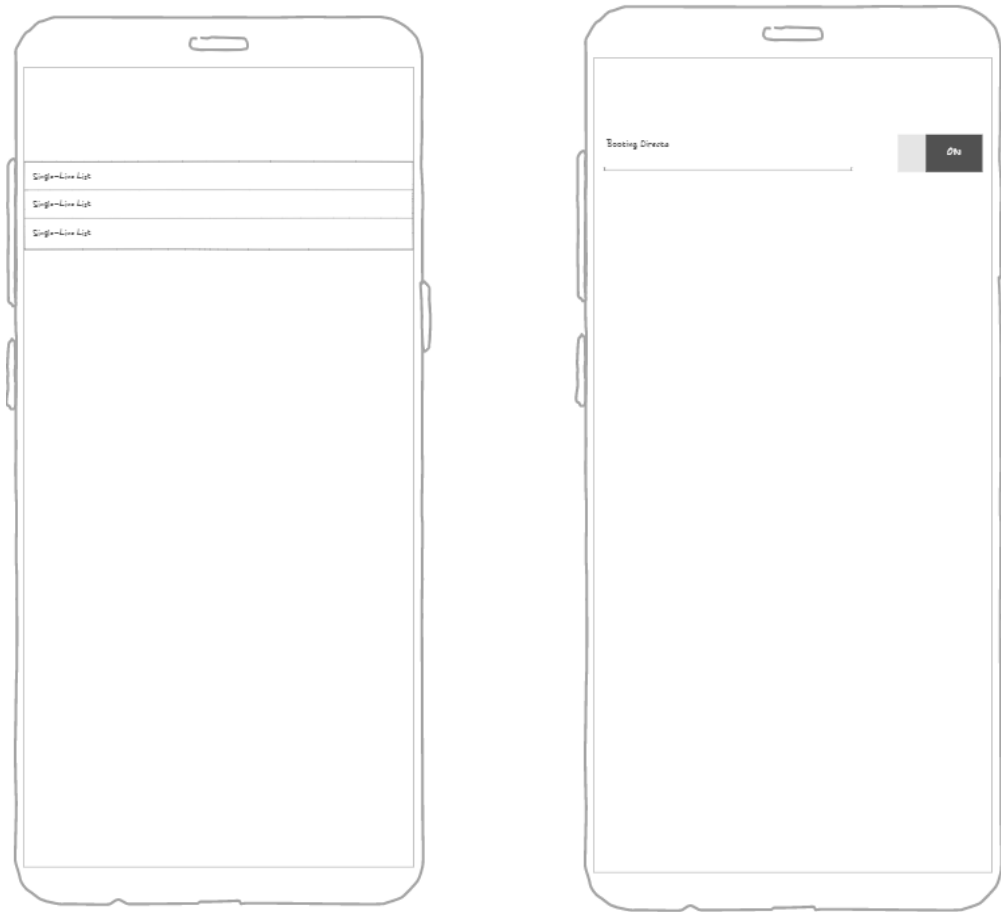


Figura 5-3 Prototipo pantalla ajustes

5.4 Diseño de la base de datos

En esta sección tratamos los aspectos de diseño relacionados con la base de datos. Los diagramas de esta sección están realizados con Visual Paradigm [9].

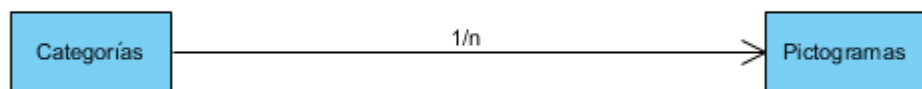


Figura 5-4 Modelo Entidad Relación

La ilustración nos muestra el esquema entidad relación de la base de datos. Estamos ante una base de datos sencilla para almacenar los pictogramas en categorías.



Figura 5-5 Tablas de la Base de Datos

En esta figura podemos ver cómo serán las tablas de la base de datos y sus atributos.

La base de datos estará almacenada en diferentes ficheros SQL, un fichero por cada categoría, en el que se encontrarán todos los pictogramas de dicha categoría. Por lo tanto, estos ficheros serán: alimentos.sql, aseo.sql, interacciones.sql, lugares.sql, objetos.sql, y personas.sql. Además, habrá un fichero para almacenar las categorías: categorías.sql.

Esto se debe a que las categorías tienen representación de pictograma y por lo tanto necesitan su propio fichero.

6 Desarrollo

6.1 General

Para la realización de este proyecto se ha utilizado el patrón de arquitectura MVC (modelo-vista-controlador).

En este capítulo se explicarán los detalles relevantes, así como las diferentes dificultades que se encontraron en el proceso de desarrollo.

En esta fase del proyecto se implementa el diseño realizado en el capítulo anterior de diseño, basado en el catálogo de requisitos Capítulo 4 – Análisis.

Como explicamos en el capítulo anterior utilizamos la herramienta Android Studio junto al SDK de Android para el desarrollo de la aplicación.

6.1.1 Modelo

Dentro del modelo distinguimos dos partes:

6.1.1.1 Base de datos

La base de datos como vimos en el capítulo anterior sobre el diseño tiene dos tablas:

- CategoriaTable: es la tabla en la que se definen las categorías.
- PictogramaTable: es la tabla en la que se definen los pictogramas.

Ambas tablas están asociadas pues cada pictograma tiene una categoría.

Dentro de la base de datos también hemos creado una clase “DatabaseAdaper” donde se crea la base de datos, actualiza y se encuentran los métodos relacionados con ella. Esta clase se encarga de gestionar la base de datos.

6.1.1.2 Monitor

El monitor es una clase que hemos creado para implementar un “BroadcastReciver” y de esta forma poder analizar las acciones del sistema.

En nuestro caso lo queremos para ver cuando el dispositivo se enciende, ya que uno de nuestros requisitos era dar la opción de que la aplicación se ejecutase de forma automática al encenderse.

Para esta labor, aparte de esta clase, hemos tenido que incluir un permiso y un elemento receiver en el archivo AndroidManifest.xml.

```

<receiver android:enabled="false"
    android:name=".model.monitor.Monitor"
    android:permission="android.permission.RECEIVE_BOOT_COMPLETED">

    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"/>
        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</receiver>

```

Figura 6-1 Receiver AndoridManifest

```

<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>

```

Figura 6-2 Permisos AndroidManifest

6.1.2 Vista

Las vistas de la aplicación son los archivos “xml” que muestran la aplicación. Básicamente son copias de las maquetas realizadas en el capítulo anterior, pero en Android Studio y con funcionalidad real. Ver capturas en el siguiente apartado.

6.1.3 Controlador

El controlador es la última parte de la aplicación y nos ayuda a relacionar las vistas con el modelo.

En esta sección se encuentran las activity:

6.1.3.1 MainActivity

La clase MainActivity controla la pantalla de inicio y muestra las categorías.

Esta clase accede a la base de datos para mostrar el listado de categorías y controla la actualización de las categorías que se muestran al avanzar en la lista en la interfaz.

Tanto esta pantalla como el resto de la aplicación se adaptan a cualquier dispositivo ya sea *Smartphone* o *Tablet* y en *portrait* o *landscape*. Aunque se recomienda un buen tamaño de pantalla para mejorar su usabilidad por parte del usuario.

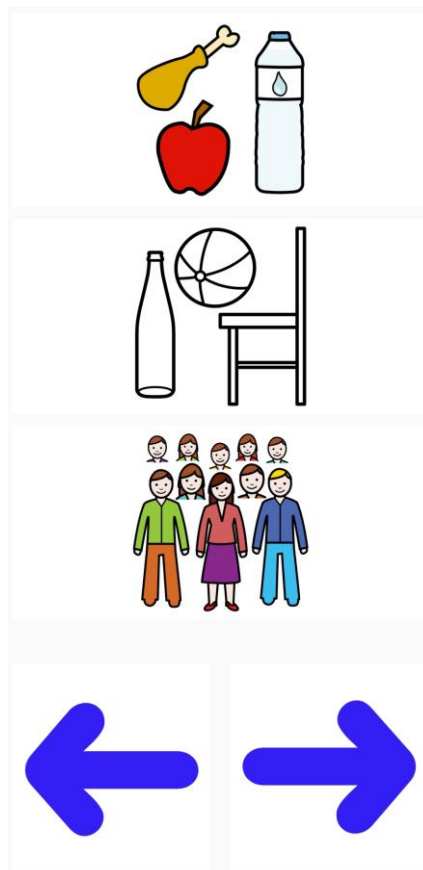


Figura 6-3 Pantalla principal Portrait

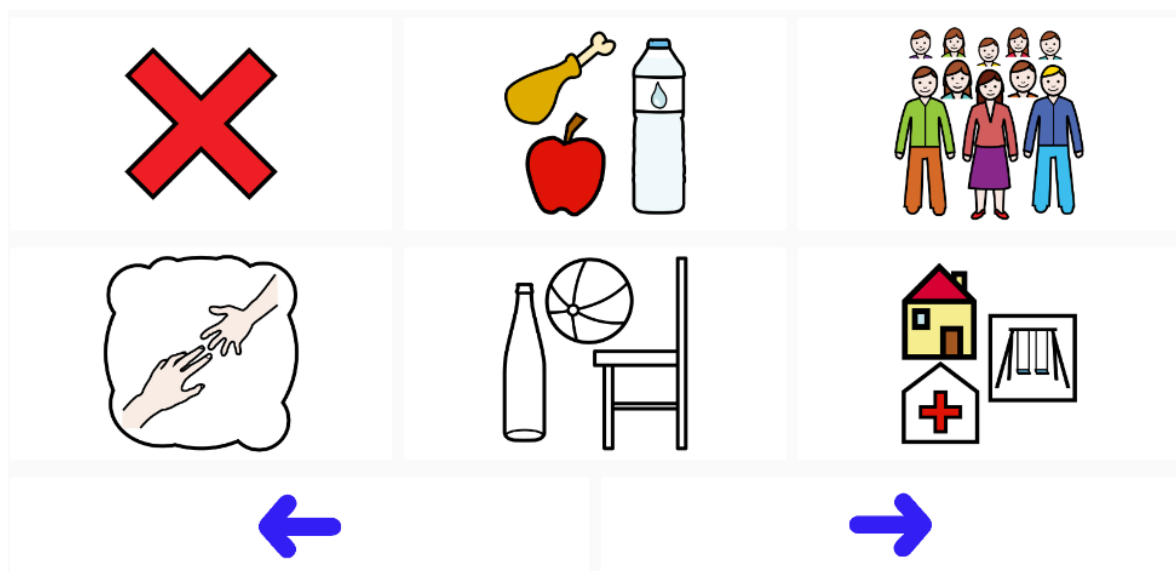


Figura 6-4 Pantalla principal Landscape

Estas imágenes son unas capturas de pantalla de mi smartphone para que se vea el resultado final de cómo se vería esta pantalla.

En el modo *landscape* al poder administrar mejor el espacio de los pictogramas en lugar de 3 categorías se muestran 4 y 2 pictogramas fijos: no y ayuda. De esta forma se facilita la comunicación.

Para conseguir esta diferencia entre ambas vistas se han desarrollado funciones que detectan la posición del dispositivo y actualizan la vista en función de su orientación.

Como explicamos en el capítulo anterior, queríamos que la pantalla se viese completa, esta acción también se controla en esta clase, de la misma forma que se controla como reducir la pantalla completa para mostrar el menú virtual inferior, las notificaciones y poder acceder a los ajustes de la aplicación con gestos en la pantalla.

```
/**
 * Metodo para poner la interfaz en modo pantalla completa
 * @param hasFocus
 */
@Override
public void onWindowFocusChanged(boolean hasFocus) {
    super.onWindowFocusChanged(hasFocus);
    if (hasFocus) {
        hideSystemUI();
    }
}

private void hideSystemUI() {
    // Enables regular immersive mode.
    // For "lean back" mode, remove SYSTEM_UI_FLAG_IMMERSIVE.
    // Or for "sticky immersive," replace it with SYSTEM_UI_FLAG_IMMERSIVE_STICKY
    View decorView = getWindow().getDecorView();
    decorView.setSystemUiVisibility(
        View.SYSTEM_UI_FLAG_IMMERSIVE
        // Set the content to appear under the system bars so that the
        // content doesn't resize when the system bars hide and show.
        | View.SYSTEM_UI_FLAG_LAYOUT_STABLE
        | View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION
        | View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN
        // Hide the nav bar and status bar
        | View.SYSTEM_UI_FLAG_HIDE_NAVIGATION
        | View.SYSTEM_UI_FLAG_FULLSCREEN);
}

// Shows the system bars by removing all the flags
// except for the ones that make the content appear under the system bars.
private void showSystemUI() {
    View decorView = getWindow().getDecorView();
    decorView.setSystemUiVisibility(
        View.SYSTEM_UI_FLAG_LAYOUT_STABLE
        | View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION
        | View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN);
}
```

Figura 6-5 Código Pantalla Completa

6.1.3.2 *PictogramaActivity*

La clase *PictogramaActivity* controla la pantalla de pictogramas asociados a una categoría y los muestra.

Esta clase accede a la base de datos para obtener el listado de pictogramas de la categoría a mostrar y controla su actualización en pantalla.

En el caso de esta *activity* al igual que en la anterior se desarrolló un menú inferior para moverse por la lista de pictogramas, pero se añadió un botón para regresar a la pantalla anterior (*MainActivity*).

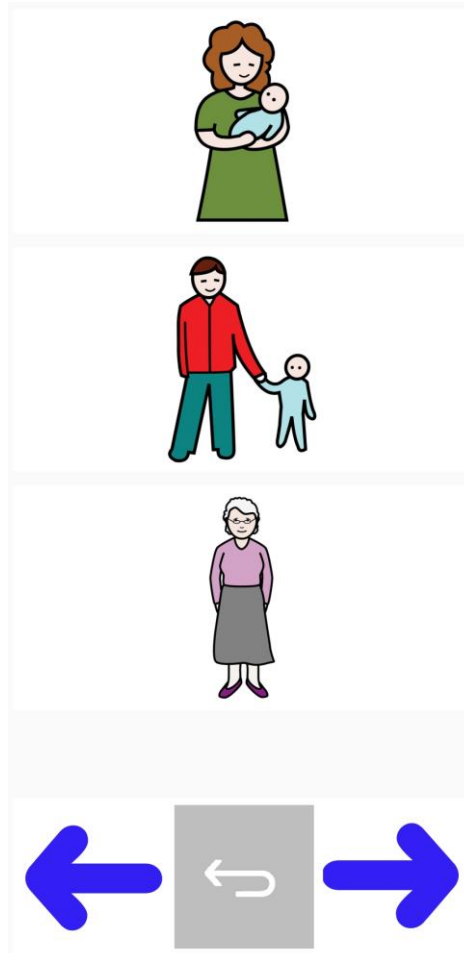


Figura 6-6 Pantalla Pictograma Portrait

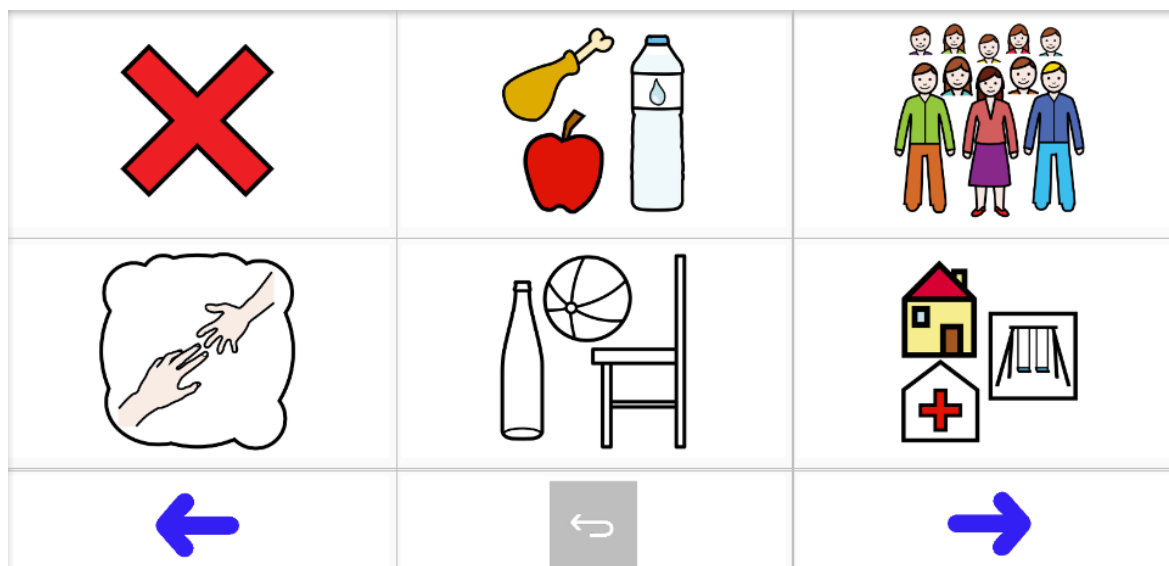


Figura 6-7 Pantalla Pictograma Landscape

Estas imágenes son unas capturas de pantalla de mi smartphone para que se vea el resultado final de cómo se vería esta pantalla.

Al igual que en MainActivity se implementa la pantalla completa, ver Figura 6-5. Y se adapta a cualquier dispositivo ya sea *Smartphone* o *Tablet* y en *portrait* o *landscape*.

6.1.3.3 SettingActivity

La clase SettingActivity es la encargada de gestionar los ajustes de la aplicación. Se ha realizado una implementación siguiendo los estándares de Android, de forma que primero tenemos un listado de ajustes, desde el cual accedemos a cada tipo de ajustes.

En nuestro caso solo hemos implementado un ajuste, para decidir si la aplicación realiza el *booting* directo o no.

La idea es ir añadiendo en futuras versiones más ajustes como podría ser añadir nuevos pictogramas, nuevas categorías, modificar los pictogramas, etc.

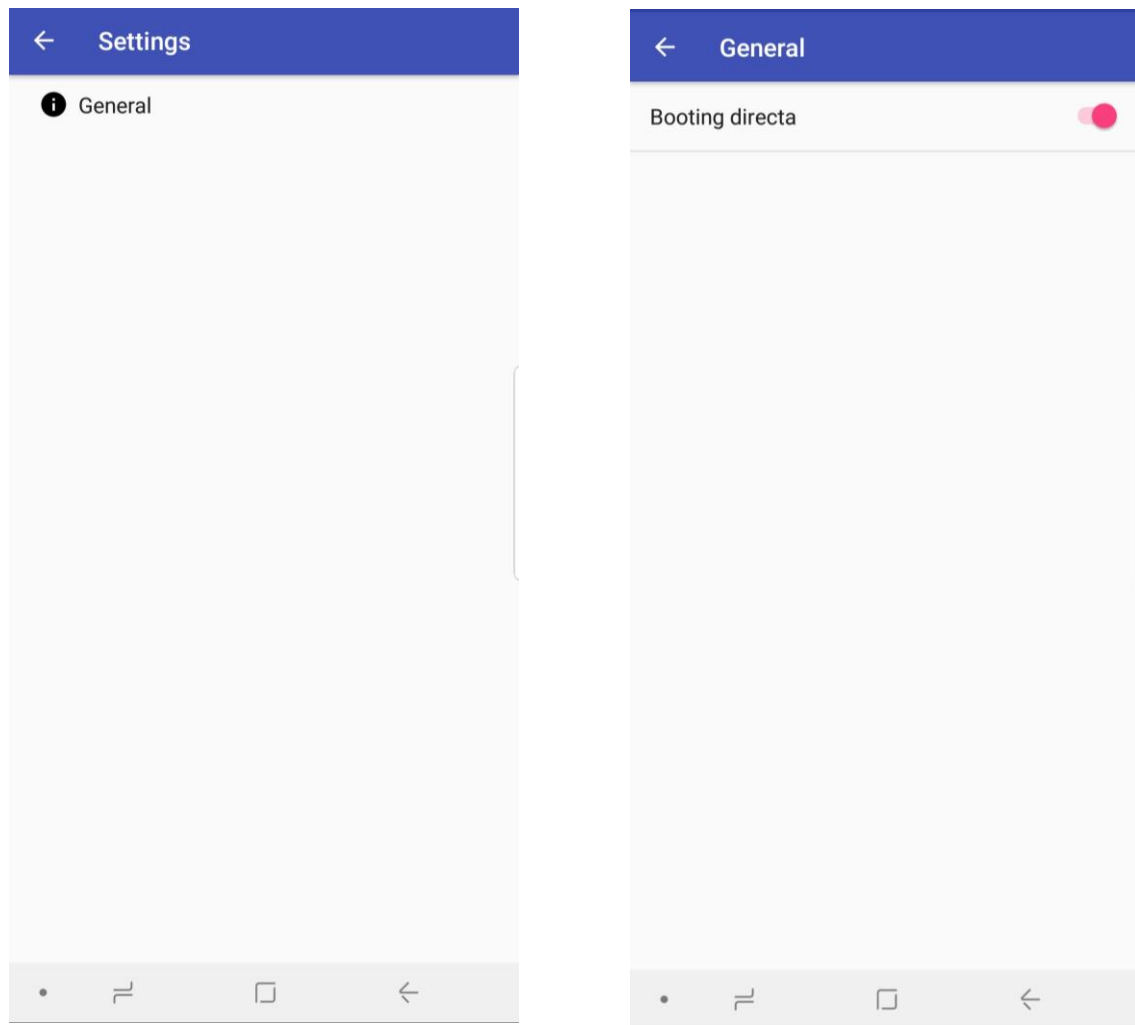


Figura 6-8 Pantalla *Settings*

Estas dos imágenes nos muestran como es el menú de ajustes, como hemos dicho esta implementado para poder ampliarlo de forma rápida y eficaz. La primera pantalla es un *activity* y la segunda un *fragment* de forma que para añadir funcionalidades se podría o bien añadir en el *fragment* si fuese un ajuste general o añadir otro *fragment* en *setting* para ajustes de otro tipo, como podría ser ajustes de usuario.

7 Pruebas

En cada paso del desarrollo se ha ido probando la aplicación tanto en emuladores creados con Android Studio como en un smartphone físico (en este caso el mío particular). De esta forma se ha podido ir verificando el correcto funcionamiento de todas las funcionalidades implementadas.

También gracias a estas pruebas, principalmente a las del smartphone físico se han podido ir tomando decisiones en el desarrollo, como en el caso del menú inferior, ya que en el emulador no se podía probar con la aplicación EVA Facial Mouse [3] y el manejo no era real.

Para comprobar tanto la facilidad de uso como la facilidad de instalación se ha pedido ayuda a familiares y amigos que han podido probar esta aplicación. De esta forma no solo pude comprobar su usabilidad, además me aportaron sus opiniones y sugerencias que me permitieron mejorar la versión final.

La aplicación será probada finalmente por dos alumnas con Pluridiscapacidad (cognitiva y motórica) por Parálisis cerebral Infantil, escolarizadas en la Unidad de Educación Especial del centro CEIP Romero Peña de La Solana (Ciudad Real). Estas pruebas empezarán en el próximo curso académico y la utilidad de la aplicación será ampliada a otros alumnos con graves discapacidades en los cuales su movilidad se limita al control de su cabeza y mirada. Ver Anexo B.

7.1 Plan de pruebas

Para probar correctamente la aplicación se implementado un plan de pruebas en el que se incluye cada funcionalidad de la aplicación:

- Prueba 1: Cambio de categorías con flecha derecha de la barra de navegación.
- Prueba 2: Cambio de categorías con flecha izquierda de la barra de navegación.
- Prueba 3: Acceso a la categoría adecuada con los botones de categorías.
- Prueba 4: Regreso al nivel de categorías con el botón de atrás de la barra de navegación.
- Prueba 5: Cambio de pictogramas dentro de una categoría con la flecha derecha de la barra de navegación.
- Prueba 6: Cambio de pictogramas dentro de una categoría con la flecha izquierda de la barra de navegación.
- Prueba 7: Reproducción del sonido adecuado con el botón de un pictograma.
- Prueba 8: Acceso al menú de ajustes con deslizamiento de pantalla hacia abajo en la parte superior del dispositivo.
- Prueba 9: *Booting* directa a la aplicación con su opción activada en los ajustes de la aplicación.
- Prueba 10: Comprobación de que no se ejecuta la aplicación en *booting* directa al desactivar la opción en ellos ajustes de la aplicación.

Finalmente existen ciertas pruebas que se hacen únicamente visualmente como que la pantalla de la aplicación se ejecuta en pantalla completa y que se elimina la barra de navegación por defecto de Android.

8 Conclusiones y trabajo futuro

8.1 Conclusiones

Este capítulo está dedicado a las conclusiones personales del proyecto, extraídas a lo largo de todo el proceso de desarrollo del mismo.

Lo primero es mencionar que se han cumplido todos los requisitos mencionados en los capítulos. El principal objetivo era crear una aplicación móvil para Android que con ayuda de la aplicación EVA Facial Mouse [3] que sirviese de comunicador para personas con discapacidad motora en los brazos y así controlar la aplicación con movimientos de cabeza. Y la aplicación está lista para ser probada en el CEIP Romero Peña de La Solana (Ciudad Real) en cuanto comience el próximo curso académico,

La aplicación desarrollada ofrece la posibilidad de seleccionar por categorías los pictogramas básicos.

Lo más complicado del proyecto era solucionar el problema configuración de ambas aplicaciones y tras varios intentos de integración se decidió crear un buen manual de instalación de la aplicación EVA Facial Mouse [3], la cual una vez instalada y configurada no se usará más y dentro de nuestra aplicación tomar las medidas para que el usuario no pueda salir, cambiar o realizar una acción en la aplicación con la cual le resulte complicado regresar a la misma.

La finalidad de la aplicación desarrollada es que el usuario de una forma muy intuitiva pueda comunicarse basándose en pictogramas y para eso necesitábamos una aplicación con un diseño muy sencillo e intuitivo, sin elementos que puedan distraer al usuario o quiten espacio en pantalla para la utilidad principal de la aplicación.

El super usuario o administrador, que será la persona encargada de instalar y ayudar a configurar ambas aplicaciones mediante gestos en la pantalla podrá acceder a los ajustes, salir de la aplicación y utilizar el dispositivo de una forma normal.

En el lado personal, he aprendido bastante sobre la utilización de la aplicación EVA Facial Mouse [3], así como de otras utilidades que tiene que no se necesitan en este proyecto, pero que podrían ser interesante para versiones futuras.

Además, he aprendido a desarrollar un proyecto de principio a fin un poco más elaborado que los realizados en las prácticas durante mis estudios. Ampliando en conocimientos sobre Android, SQL y la gestión del desarrollo de un proyecto.

8.2 Trabajo futuro

Dentro del trabajo futuro en este proyecto pueden destacarse varias opciones:

La primera de ellas sería la mejora de ajustes del usuario, para que el super usuario pueda añadir nuevos pictogramas y categorías a la aplicación sin necesidad de utilizar ficheros *sql*.

Un aspecto interesante que mejorar sería crear un algoritmo que nos muestre los pictogramas y categorías según nuestro uso de la aplicación. Una primera aproximación sería que el super usuario pudiera reorganizar a gusto del usuario las listas.

Por otro lado, Android es un sistema operativo que se actualiza continuamente, por lo que debería programarse una tarea de mantenimiento para estudiar nuevas versiones y adaptar el proyecto, evitando de este modo comportamientos erróneos de la aplicación en futuras versiones de Android.

Referencias

- [1] DSLab UAM, Quiero Decirte... (Versión 1.11) [Aplicación Móvil] Descargado de: <https://play.google.com/store/apps/details?id=dslab.quierodecirte&hl=es> 419
- [2] ARASAAC Portal Aragonés de la Comunicación Aumentativa y Alternativa: <http://www.arasaac.org/index.php>
- [3] Fundación Vodafone España, EVA Facial Mouse (Versión 3.0.1) [Aplicación Móvil] Descargado de: https://play.google.com/store/apps/details?id=com.crea_si.eviacam.service&hl=es
- [4] Irisbond Descubre todo El Potencial De tu Mirada: <https://www.irisbond.com/>
- [5] Speak for Yourself LLC, Speak for Yourself (Versión 3.6) [Aplicación Móvil] Descargado de: <https://apps.apple.com/us/app/speak-for-yourself/id482508198/?platform=iphone>
- [6] CoughDrop AAC: <https://www.coughdrop.com/>
- [7] Eva Facial mouse, GitHub: https://github.com/cmauri/eva_facial_mouse
- [8] NinjaMock: <https://ninjamock.com/home/index?v=2>
- [9] Visual Paradigm: <https://www.visual-paradigm.com/tutorials/define-custom-implementations-for-orm.jsp>
- [10] Android Studio: <https://developer.android.com/studio>

Glosario

- **Smartphone:** Anglicismo. Teléfono móvil construido sobre una plataforma informática móvil, con capacidad para almacenar datos y realizar actividades asemejándose a una minicomputadora.
- **Android:** Sistema operativo basado en Linux para dispositivos móviles.
- **Linux:** Combinación de *kernel* libre con sistema GNU.
- **GNU:** Sistema operativo desarrollado por GNU, formado en su totalidad por software libre.
- **Kernel:** Núcleo, software que constituye una parte fundamental de un sistema operativo.
- **Activity:** Cada una de las pantallas de la aplicación.
- **ARASAAC:** Portal aragonés de la Comunicación Aumentativa y alternativa.
- **UML:** *Unified Modeling Language*, es el lenguaje de modelado de sistemas de software más conocido.
- **SysML:** *Sustems Modeling Language*, es un lenguaje de especificación de sistemas.
- **IDE:** Entorno de desarrollo integrado, es una aplicación informática que proporciona servicios integrales para facilitar el desarrollo de software.
- **Gradle:** es un sistema de automatización de construcción de código abierto.
- **APK:** es una extensión de archivo, en forma de paquete para el sistema operativo Android.
- **GitHub:** es una plataforma para alojar proyectos utilizando el sistema de control de versiones Git.
- **Git:** es un software de control de versiones.
- **SQL:** *Structured Query Language*, es un lenguaje de dominio específico utilizado en programación, diseñado para administrar, y recuperar información.

Anexos

A Manual de instalación

En este manual vamos a aprender a instalar y configurar todo lo necesario para poder utilizar la aplicación con éxito.

A1. EVA Facial Mouse

Para el uso de nuestra aplicación necesitamos instalar y configurar primero la aplicación “EVA Facial Mouse”, para ello nos dirigimos a Google Play, buscamos la aplicación y la instalamos o accedemos a través del siguiente enlace: https://play.google.com/store/apps/details?id=com.crea_si.eviacam.service&hl=es.

Una vez instalada EVA Facial Mouse accedemos a ella y nos pedirá distintos permisos para la utilización de la aplicación:

- Permiso para mostrar contenido encima de otras aplicaciones, mostrar menús y otros elementos.
- Permiso para acceder a la cámara frontal para el seguimiento de la cara.

Además, nos pedirá que aceptemos los términos y condiciones de uso de la aplicación y que activemos el servicio de ajustes de accesibilidad.

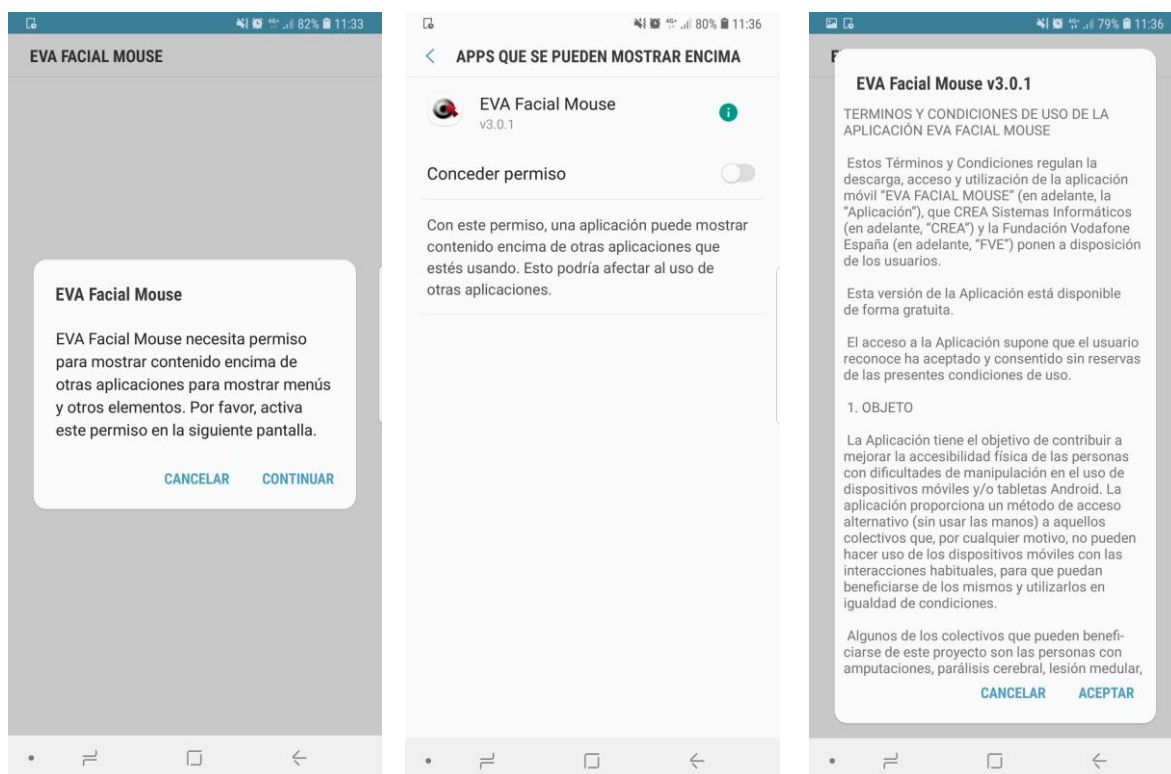


Figura 0-1 Instalación EVA Facial Mouse 1

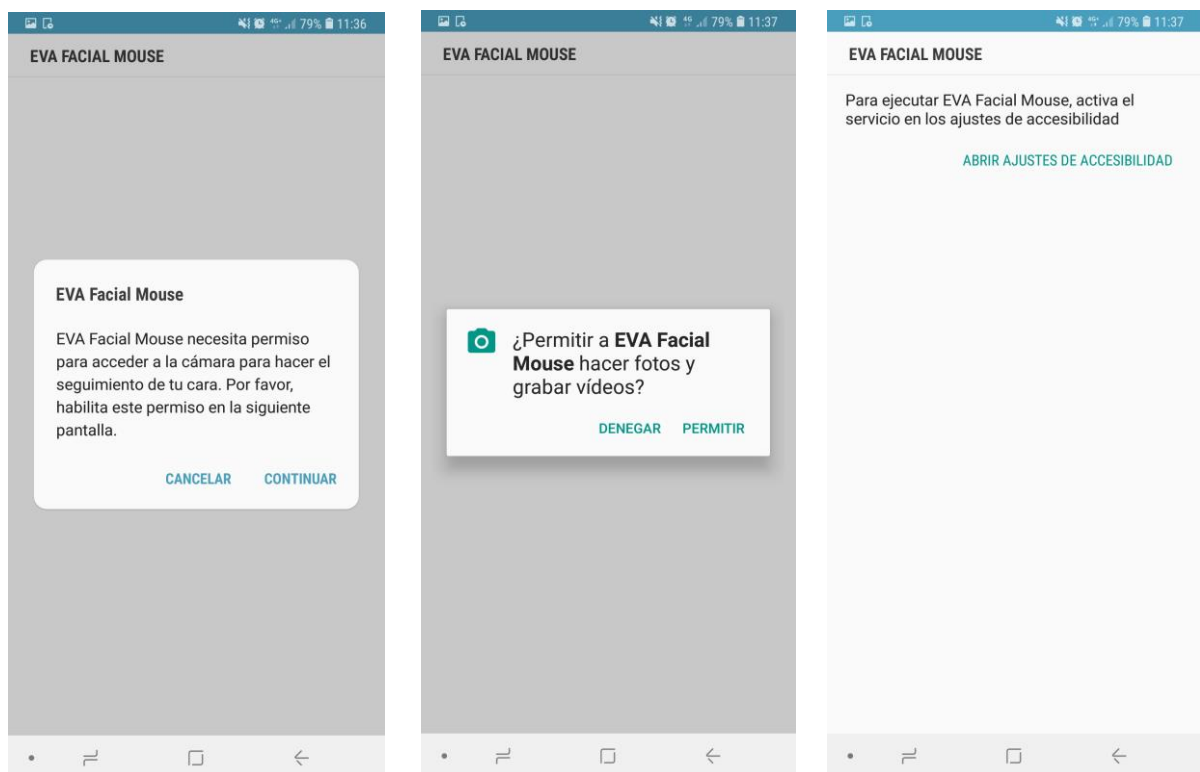


Figura 0-2 Instalación EVA Facial Mouse 2

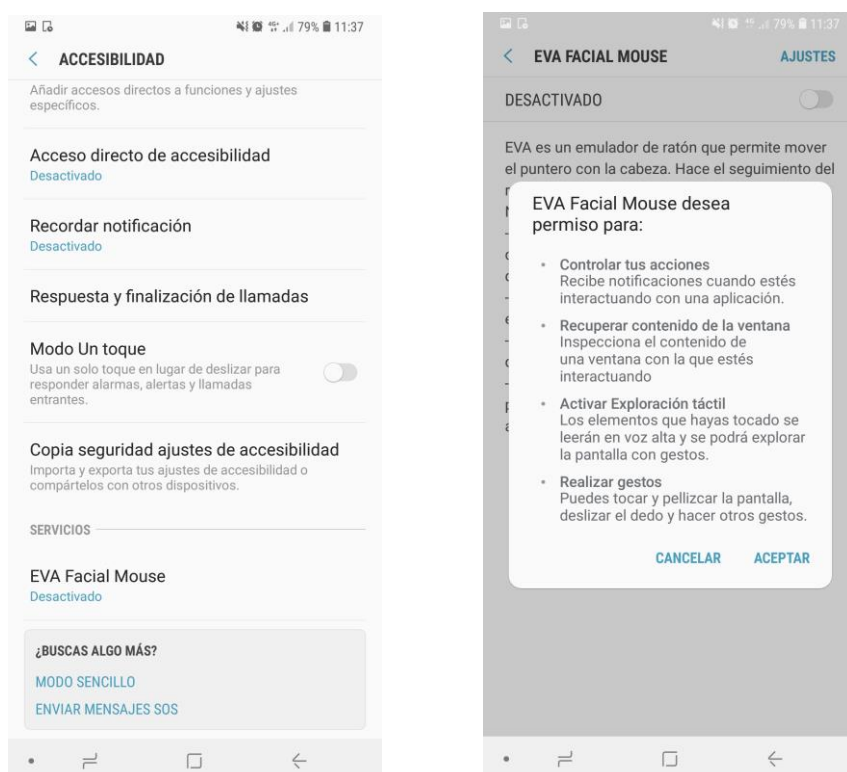


Figura 0-3 Instalación EVA Facial Mouse 3

Tras este proceso nos mostrara una pantalla con el inicio del asistente o la opción de omitirlo. Mi recomendación es que realices el asistente en caso de ser la primera vez que utilizas la aplicación.



Figura 0-4 Instalación EVA Facial Mouse 4

Nosotros vamos a seguir el manual para descubrir todas las posibilidades de esta aplicación y configurarla correctamente. Para ello nos iremos ayudando de imágenes.

- 1) El principal uso de esta aplicación es controlar el dispositivo con movimientos de la cabeza. Al iniciar el asistente nos aparecerá un cuadrado pequeño en la parte superior de la pantalla y tendremos que ajustar la distancia al móvil hasta que se ponga de color verde. En este momento el asistente nos indicara que el rostro ha sido detectado.



Figura 0-5 Instalación EVA Facial Mouse 5

- 2) Una vez detectado nuestro rostro ajustamos la sensibilidad de movimiento horizontal y vertical para con el puntero que aparece en pantalla poder llegar a cualquier lugar de la pantalla. Lleva el puntero a las esquinas del dispositivo. Una vez conseguido ya podemos movernos por el dispositivo con el rostro.

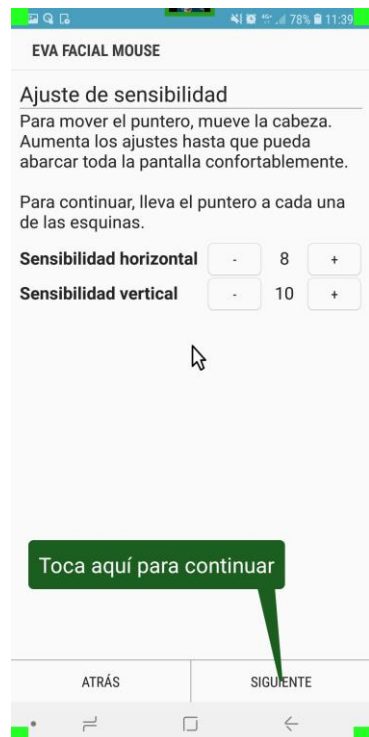


Figura 0-6 Instalación EVA Facial Mouse 4

- 3) Ajustamos el temblor del puntero, para ello nos dan cuatro opciones de suavizado de movimiento.

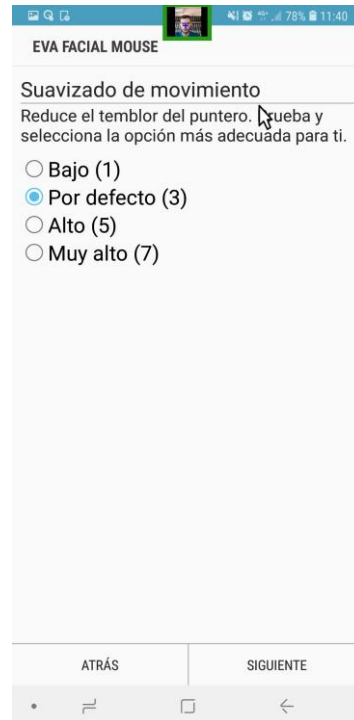


Figura 0-7 Instalación EVA Facial Mouse 7

- 4) Para realizar pulsaciones en iconos, botones... será necesario detener el puntero. Por eso nos permiten ajustar cuando detener el puntero según el umbral de movimiento. En este caso nos dan cuatro opciones también.

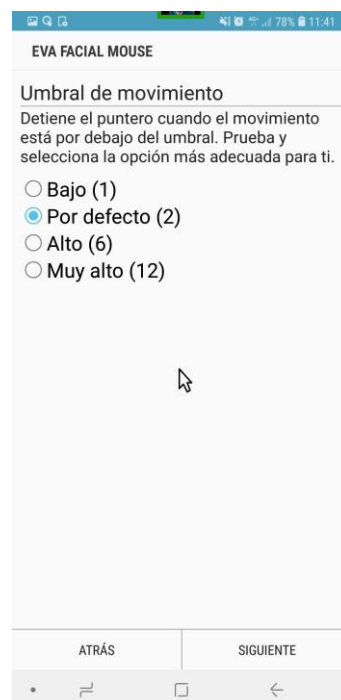


Figura 0-8 Instalación EVA Facial Mouse 8

5) El último ajuste es la aceleración del puntero. Volvemos a tener cuatro opciones.

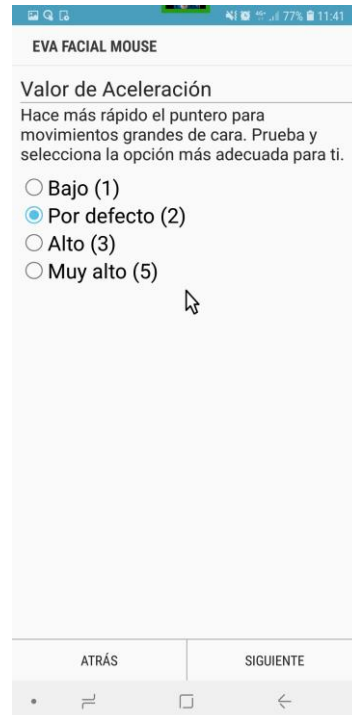


Figura 0-9 Instalación EVA Facial Mouse 9

6) Pulsación: Una vez que tenemos los movimientos del puntero ajustados, nos toca practicar las distintas opciones que tenemos para navegar por el dispositivo.

La primera es pulsar un botón, para ello detenemos el puntero encima del botón.



Figura 0-10 Instalación EVA Facial Mouse 10

7) Pulsaciones largas; lo que para nosotros sería mantener pulsado.

Para este tipo de acciones necesitaremos acceder a un menú especial donde se nos incidan las distintas acciones que podemos realizar.

En este caso accedemos pulsamos la opción de pulsación larga y la siguiente vez que pulsemos un icono será con una pulsación larga.

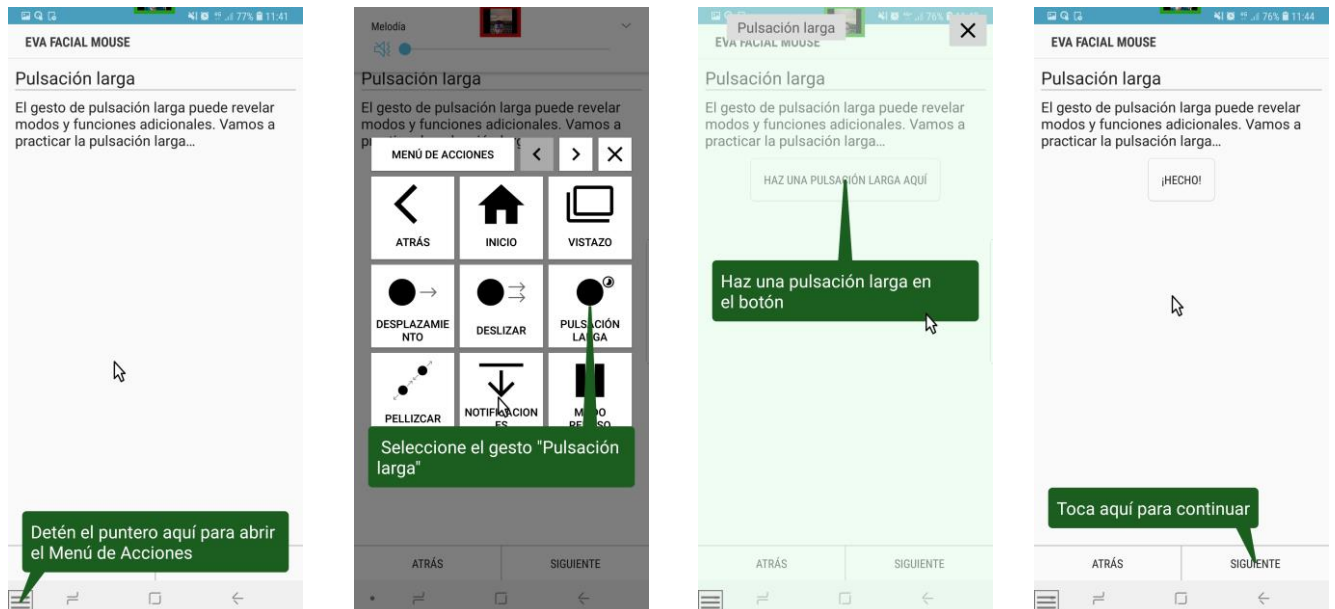


Figura 0-11 Instalación EVA Facial Mouse 11

8) Desplazar la pantalla vertical u horizontalmente.

Para ello volvemos al menú de acciones y seleccionamos desplazamiento. La siguiente vez que pulsemos en la pantalla nos aparecerá una flecha y según donde la estiramos y pulsemos la pantalla se desplazará en consecuencia.

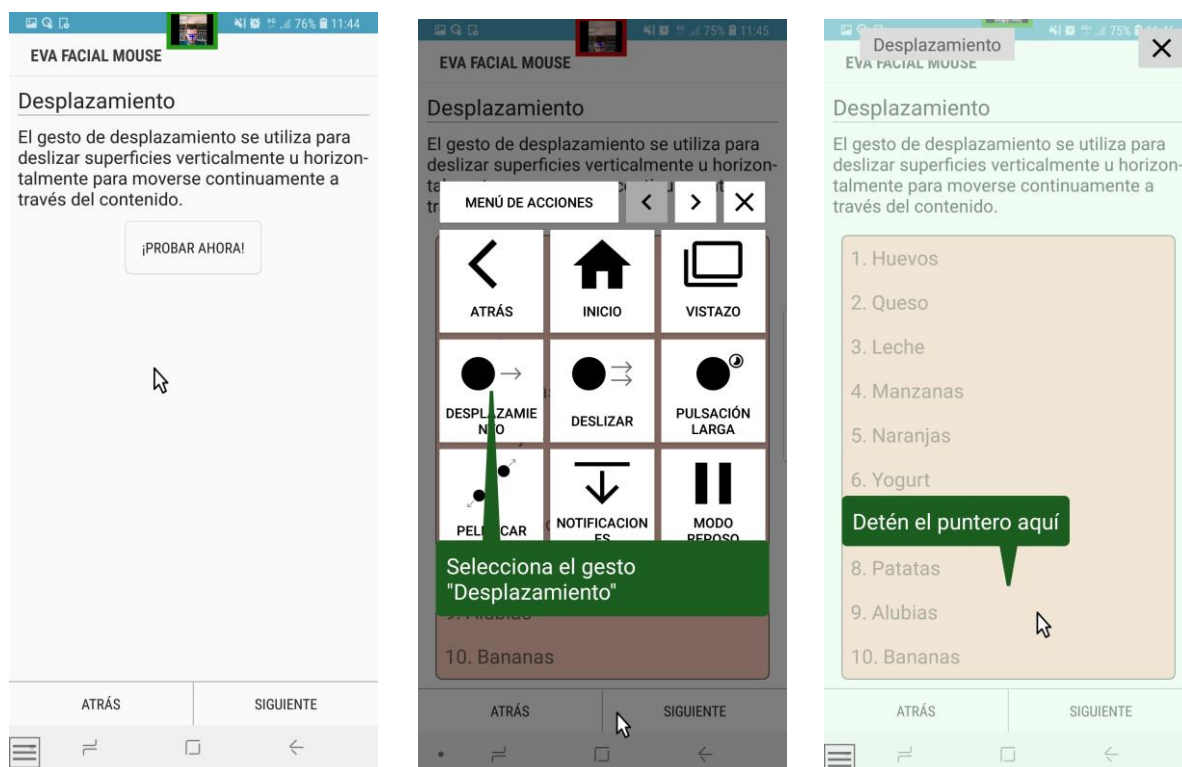


Figura 0-12 Instalación EVA Facial Mouse 12

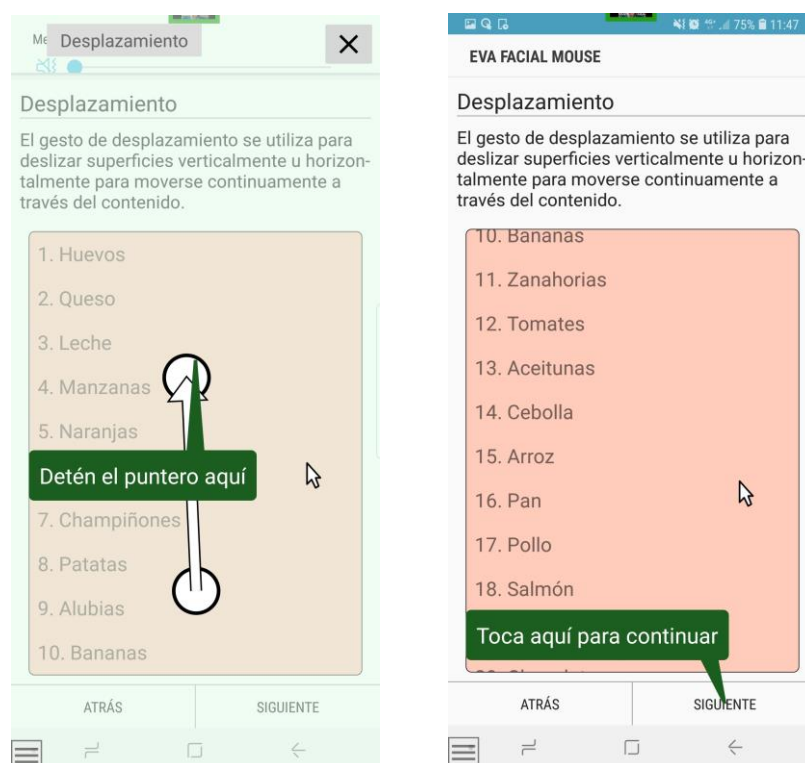


Figura 0-13 Instalación EVA Facial Mouse 13

Otra opción para desplazarnos es con unos botones adicionales que añade la aplicación, de forma opcional, pero la aplicación la considera en desuso desde Android 7, ya que su funcionalidad es inconsistente.

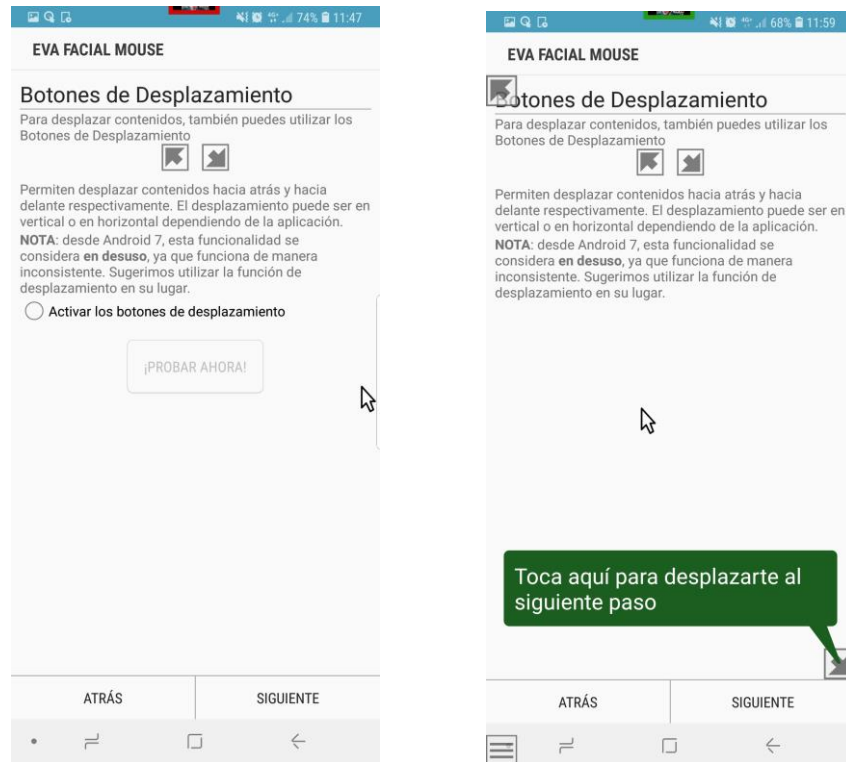


Figura 0-14 Instalación EVA Facial Mouse 14

9) Deslizar: esta acción es parecida a la anterior, pero es más rápida.



Figura 0-15 Instalación EVA Facial Mouse 15

- 10) Zoom: para ello vamos al menú de acciones, pulsamos la acción zoom. La siguiente pulsación que hagamos en la pantalla hará que nos aparezcan dos flechas en sentido opuesto y que crecen según alejemos la siguiente pulsación para realizar el zoom.

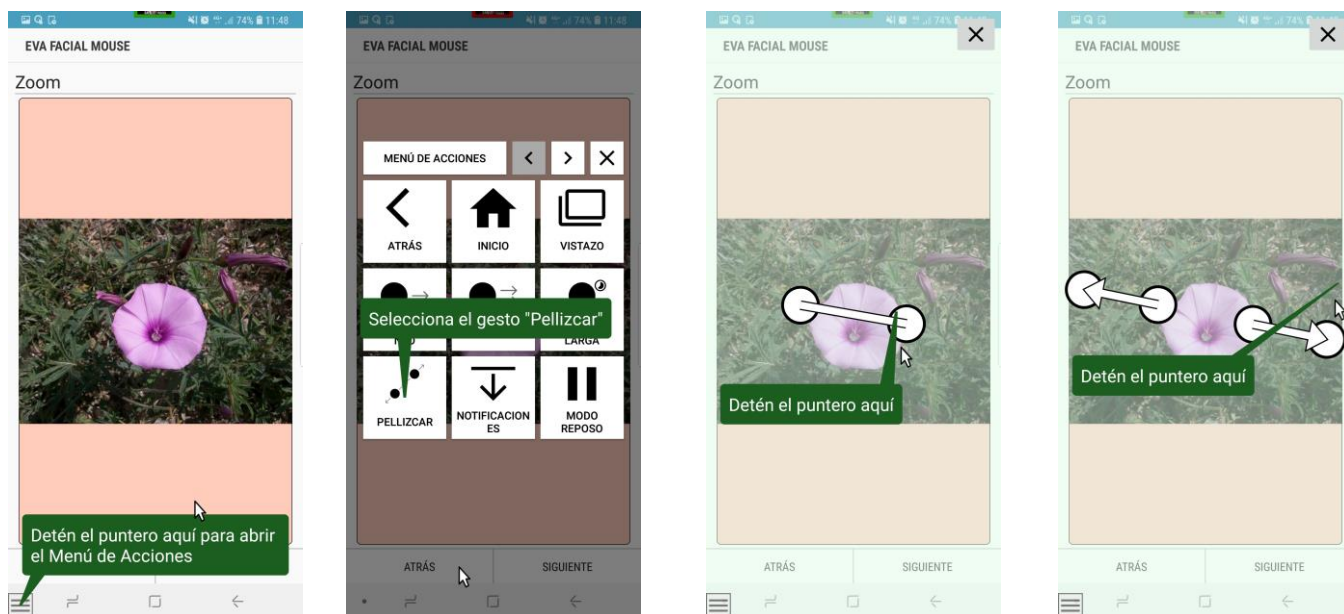


Figura 0-16 Instalación EVA Facial Mouse 16

- 11) Doble toque: esta acción se encuentra en la segunda página del menú de acciones. Una vez pulsada la siguiente pulsación que realicemos será un doble toque. Un ejemplo de uso sería para ampliar una imagen en la galería.

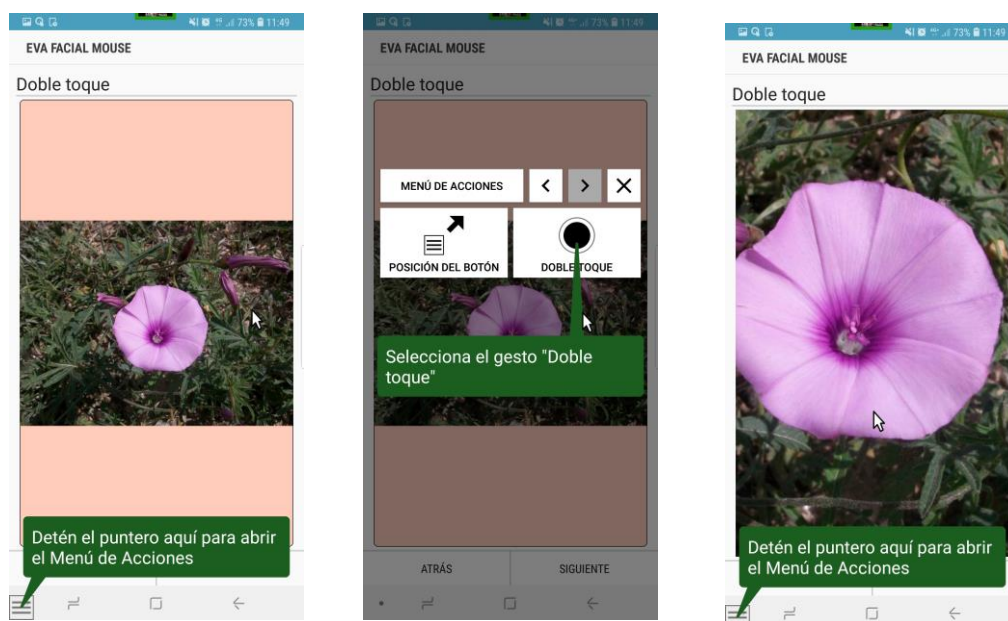


Figura 0-17 Instalación EVA Facial Mouse 17

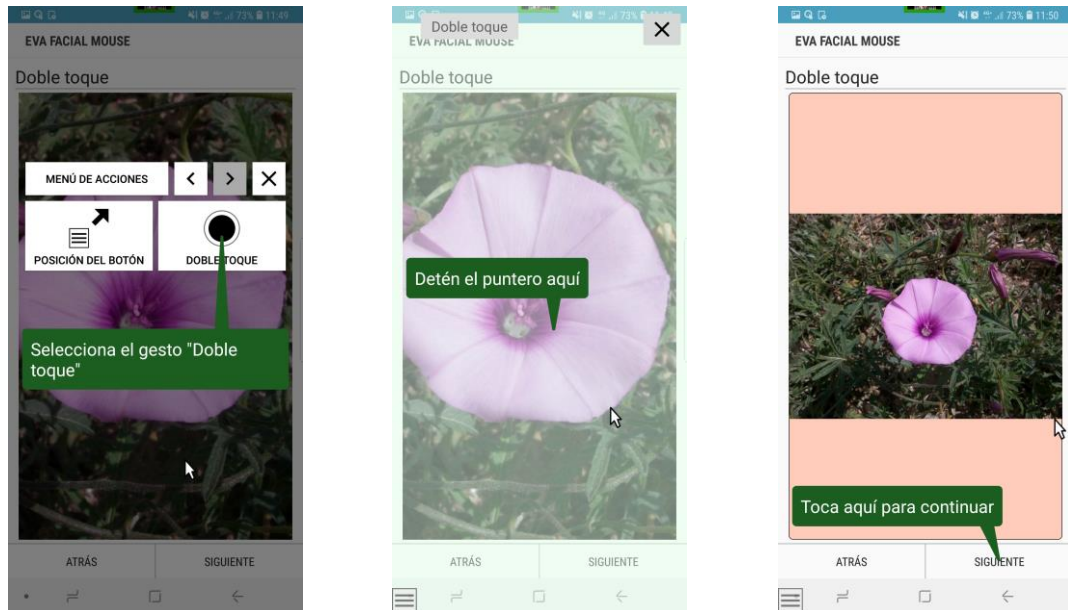


Figura 0-18 Instalación EVA Facial Mouse 18

- 12) Dentro del menú de acciones también se encuentran: atrás, inicio, vistazo, notificaciones y conmutar el modo reposo para desactivar el clic automático mientras lees o ves un vídeo.

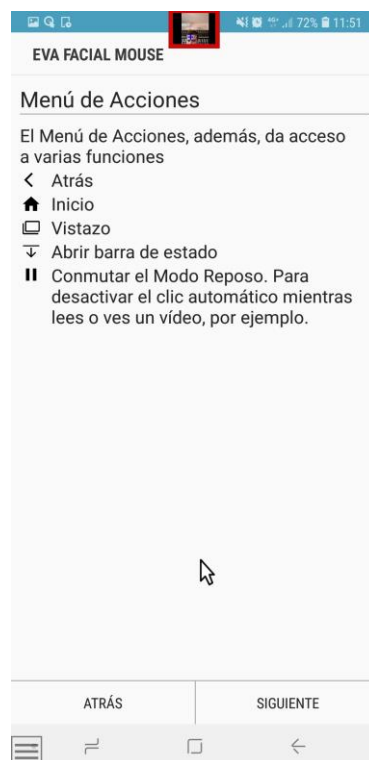


Figura 0-19 Instalación EVA Facial Mouse 19

- 13) El menú de acciones tiene dos posiciones, una en un botón el cual al ser pulsado nos muestra el menú, y otro en una barra lateral, con iconos más pequeños.



Figura 0-20 Instalación EVA Facial Mouse 20

Ya hemos terminado el asistente y por lo tanto ajustado y aprendido a usar toda la aplicación.

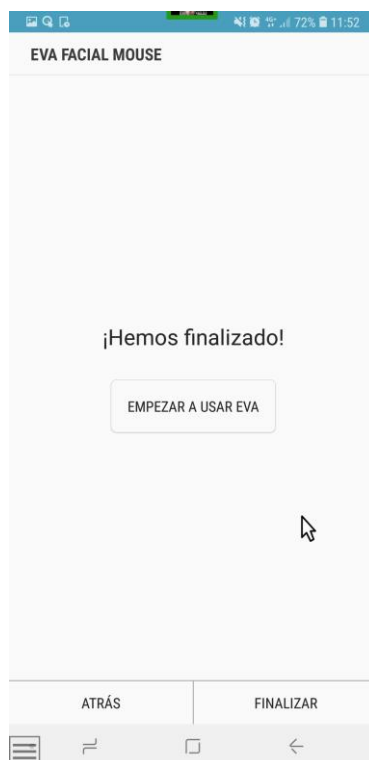


Figura 0-21 Instalación EVA Facial Mouse 21

De todas formas, al abrir la aplicación podemos acceder a los ajustes para modificarlos si fuese necesario posteriormente y volver a ver el asistente si lo deseásemos.

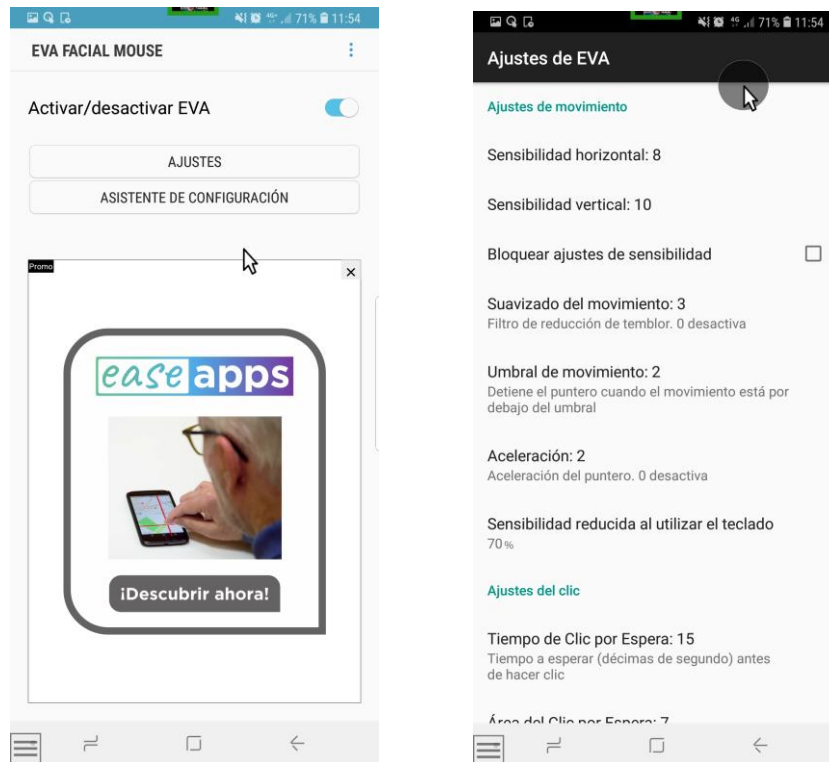


Figura 0-22 Instalación EVA Facial Mouse 22

Dentro de todas estas acciones que hemos visto, para nuestra aplicación y el uso que deseamos actualmente, solo es necesario el movimiento del rostro y la pulsación. Esto se debe a que queremos que la aplicación sea lo más intuitiva y sencilla posible. Ya que no se va a utilizar ninguna acción más podemos quitar el menú de acciones desde los ajustes de la aplicación EVA Facial Mouse.

A2. TFG

Nuestra aplicación aún no se encuentra en Google Play, por lo que su instalación es algo más complicada.

Para ello tenemos que descargar la APK del proyecto e instalarla en el sistema.

Si nunca hemos instalado un APK tenemos que conocer el proceso ya que es necesario habilitar esta instalación en el sistema (Ajustes -> Seguridad -> Orígenes desconocidos).

Pero, en breves estará incluida y su instalación será muy sencilla.

Una vez instalada lo único a tener en cuenta por parte del super usuario es la forma de acceder a los ajustes que es deslizando desde la parte superior o inferior de la pantalla el dedo. De esta forma la aplicación saldrá del modo pantalla completa.

B Carta CEIP Romero Peña de La Solana



Consejería de Educación y Ciencia
C. E. I. P. E. "Romero Peña"
San Quintín, 17 13240 LA SOLANA (C. Real)
Teléfono: 928631594, Email: 13202939.ce@edu.lccm.es

Juana Domínguez Fernández-Arroyo, funcionaria de carrera en la especialidad de Audición y Lenguaje de la Consejería de Educación, Cultura y Deportes de la Junta de Comunidades de Castilla-La Mancha, con número de registro de personal: 7073356946A0597 y Josefa García Ariza, funcionaria de carrera en la especialidad de Pedagogía Terapéutica de la Consejería de Educación, Cultura y Deportes de la Junta de Comunidades de Castilla-La Mancha, con número de registro de personal: 2423471624A0597, docentes y miembros del Equipo de Orientación y Apoyo del CEIP Romero Peña de La Solana (Ciudad Real), hacen constar que colaboran con D. Eduardo Boemo, Profesor Titular de la Universidad Autónoma de Madrid, en el desarrollo de un comunicador para tabletas Android que facilita y potencia la comunicación espontánea mediante imágenes y sonidos, permitiendo desarrollar comportamientos interactivos intencionales autónomos.

La app desarrollada por Guillermo Díaz permitirá en su aplicación realizar demandas espontáneas mediante la tableta a través de pictogramas que se convierten en voz, utilizando la cabeza y la mirada para controlar el comunicador.

El comunicador se aplicará con dos alumnas con Pluridiscapacidad (cognitiva y motórica) por Parálisis Cerebral Infantil, escolarizadas en la Unidad de Educación Especial de dicho centro; se trata de alumnas que no poseen lenguaje oral, con una capacidad cognitiva muy baja y graves dificultades motoras que les incapacita para utilizar cualquier comunicador por control manual.

Hasta el momento, este tipo de alumnado no posee ningún sistema alternativo de comunicación debido al alto precio de los existentes en el mercado y la imposibilidad de su adquisición por parte de la familia o por el centro público en el que están escolarizadas. Este comunicador fomentará una comunicación funcional y significativa adaptándose a las capacidades tanto cognitivas como motoras de los usuarios.



Consejería de Educación y Ciencia
C. E. I. P. E. "Romero Peña"
San Quintín, 17 13040 LA SOLANA (C. Real)
Teléfono: 928631594, Email: 13002839.ce@edu.lcman.es

Las pruebas comenzarán con estas dos alumnas el próximo curso y se prevé ampliar su implantación a otros alumnos con graves discapacidades en los cuales su movilidad se limita al control de su cabeza y mirada.

La Solana, a 12 de Junio de 2020.

Fdo. Juana Domínguez Fernández-Arroyo.

Fdo. Josefa García Ariza.

